BS EN 60848:2013

BSI Standards Publication

GRAFCET specification language for sequential function charts

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 60848:2013. It is identical to IEC 60848:2013. It supersedes BS EN 60848:2002, which will be withdrawn on 3 April 2016.

The UK participation in its preparation was entrusted to Technical Committee GEL/3, Documentation and graphical symbols.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2013

ISBN 978 0 580 74570 6
ICS 01.100.25; 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2013.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 60848

June 2013

ICS 29.020

Supersedes EN 60848:2002

English version

GRAFCET specification language for sequential function charts
(IEC 60848:2013)

Langage de spécification GRAFCET pour diagrammes fonctionnels en séquence (CEI 60848:2013)

GRAFCET, Spezifikationssprache für Funktionspläne der Ablaufsteuerung (IEC 60848:2013)

This European Standard was approved by CENELEC on 2013-04-03. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Ref. No. EN 60848:2013 E

Foreword

The text of document 3/1135/FDIS, future edition 3 of IEC 60848, prepared by SC 3B “Documentation” of IEC/TC 3 “Information structures, documentation and graphical symbols" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 60848:2013.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2014-01-03
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2016-04-03

This document supersedes EN 60848:2002.

EN 60848:2013 includes the following significant technical changes with respect to EN 60848:2002:

This edition constitutes a global technical revision with the extended definition of the concept of variables introducing: internal variable, input variable and output variable.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 60848:2013 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following note has to be added for the standard indicated:

IEC 61131-3:2003	NOTE	Harmonised as EN 61131-3:2003 (not modified).

CONTENTS

INTRODUCTION

This International Standard is mainly aimed at people such as design engineers, maintenance engineers, etc., who need to specify the behaviour of a system, e.g. the control and command of an automation system, safety component, etc. This specification language should also serve as a communication means between designers and users of automated systems.

GRAFCET SPECIFICATION LANGUAGE FOR SEQUENTIAL FUNCTION CHARTS

1 Scope

This International Standard defines the GRAFCET[1] specification language for the functional description of the behaviour of the sequential part of a control system.

This standard specifies the symbols and rules for the graphical representation of this language, as well as for its interpretation.

This standard has been prepared for automated production systems of industrial applications. However, no particular area of application is excluded.

Methods of development of a specification that makes use of GRAFCET are beyond the scope of this standard. One method is for example the "SFC language" specified in IEC 61131-3, which defines a set of programming languages for programmable controllers.

NOTE See Annex C for further information on the relations between IEC 60848 and implementation languages such as the SFC of IEC 61131-3.

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.

(void)

3 Terms and definitions

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

NOTE The definitions of the terms in 3.1 apply only in the context of the GRAFCET specification language.

3.1 Terms in the GRAFCET

3.1.1
action
GRAFCET language element associated with a step, indicating an activity to be performed on output or internal variables

3.1.2
directed link
GRAFCET language element indicating the evolution paths between steps by connecting steps to transitions and transitions to steps

3.1.3
grafcet chart
function chart using the GRAFCET specification language

---

1 GRAFCET: GRAphe Fonctionnel de Commande Etape Transition.

Note 1 to entry: The “grafcet chart” can, in short form, be called “grafcet”.

3.1.4
input event
event characterized by the change of at least one value of all input variables of the sequential part of the system

3.1.5
internal event
event characterized by an input event associated with the situation of the sequential part of the system

3.1.6
interpretation
part of the GRAFCET specification language enabling the linkage of:

– the input variables and the structure, by the means of the transition-condition; and
– the output variables and the structure, by the means of the actions

3.1.7
situation
state of the system described by the GRAFCET specification language and characterized by the active steps at a given instant

3.1.8
step
GRAFCET language element used for the definition of the state of the sequential part of the system

Note 1 to entry: A step can be active or inactive.

Note 2 to entry: The set of active steps represents the situation of the system.

3.1.9
transient evolution
evolution characterized by the clearing of several successive transitions on the occurrence of a single input event

3.1.10
transition
GRAFCET language element indicating a possible evolution of the activity between two or more steps

Note 1 to entry: The possible evolution is realised by clearing the transition.

3.1.11
transition-condition
GRAFCET language element associated with a transition indicating the result of a boolean expression

Note 1 to entry: The transition-condition can be either true or false.

3.1.12
variable
scalar quantity defined by its name and Boolean, numeric value

3.1.13
input variable
variable which may influence the behaviour described by the grafcet chart

EXAMPLE Boolean variable indicating the violation of a temperature limit.

Note 1 to entry: The variable may belong to the environment or to some other system component.

3.1.14
output variable
variable which may be influenced by the behaviour described by the grafcet chart

EXAMPLE Setpoint of a PID-controller.

Note 1 to entry: The variable may belong to the environment or to some other system component.

3.1.15
internal variable
variable used inside the grafcet chart and invisible for other system components and the environment

EXAMPLES Step variable X* (symbol 2.1 of Table 1), step duration T* (symbol 2.2 of Table 1), loop counter within a grafcet chart.

3.2 Terms, general purpose

3.2.1
chart
graph
graphical presentation describing the behaviour of a system, for example the relations between two or more variable quantities, operations or states

3.2.2
structure
part of the GRAFCET specification language enabling the description of the possible evolution between situations

3.2.3
system
set of interrelated elements considered in a defined context as a whole and separated from their environment

Note 1 to entry: Such elements may be material objects and concepts as well as their results (e.g. forms of organisation, mathematical methods, programming languages).

Note 2 to entry: The system is considered to be separated from the environment and from the other external systems by an imaginary surface, which cuts the links between them and the system.

Note 3 to entry: The language GRAFCET can be used to describe the logical behaviour of any kind of system.

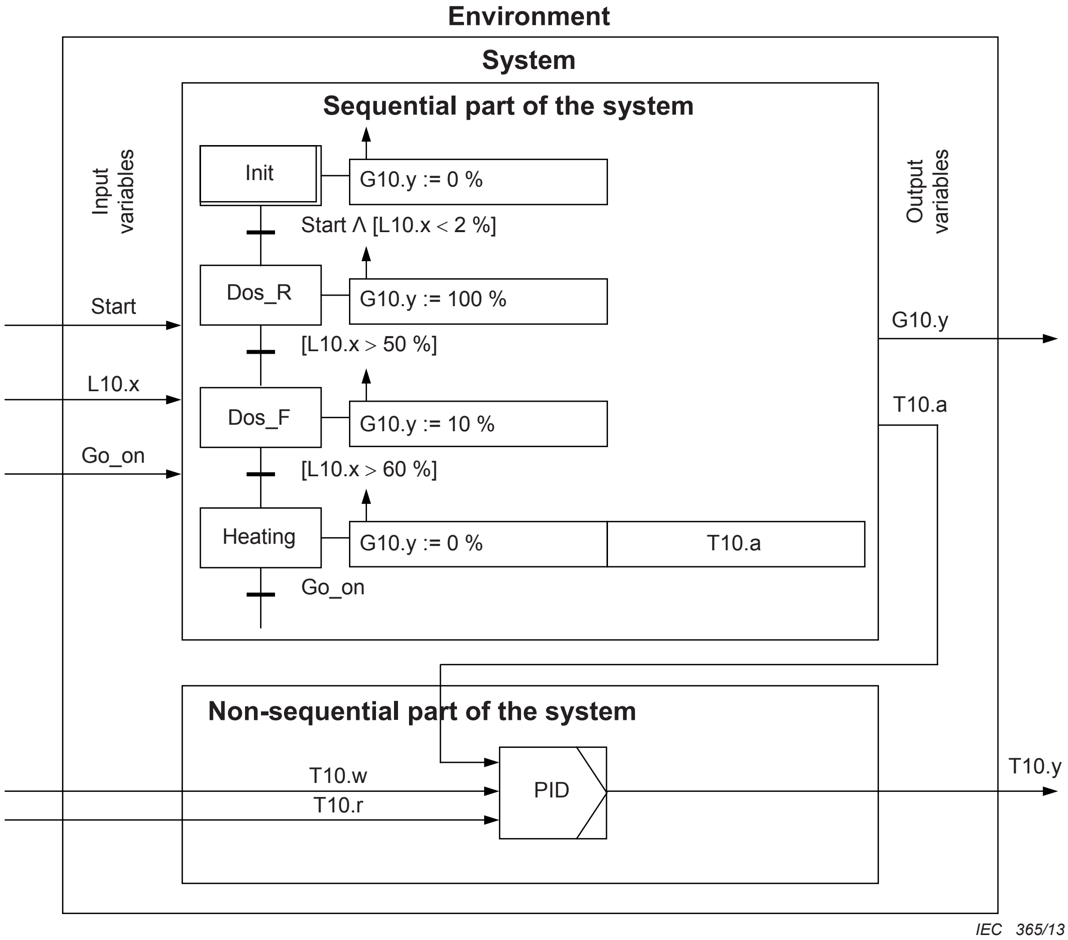
[SOURCE: IEC 60050-351:1998, 351-11-01]

4 General principles

4.1 Context

The implementation of an automated system requires, in particular, a description relating cause and effect. To do this, the logical aspect of the desired behaviour of the system will be described.

The sequential part of the system is the logical aspect of this physical system (see Figure 1). The behaviour indicates the way in which the output variables depend on the input variables. The object of the grafcet chart is to specify the behaviour of the sequential part of the systems.



L10.x tank level

T10.a temperature loop – automatic mode

T10.r temperature loop – measured value

G10.y dosing valve – position

T10.w temperature loop – setpoint

T10.y temperature loop – manipulated value

Figure 1 – Graphical representation of the sequential part of a system

4.2 GRAFCET, a behaviour specification language

The GRAFCET specification language enables a grafcet chart to be created showing the expected behaviour of a given sequential system. This language is characterized mainly by its graphic elements, which, associated with an alphanumerical expression of variables, provides a synthetic representation of the behaviour, based on an indirect description of the situation of the system.

The behaviour description on states is the following: the "monomarked" states correspond to the situations of the grafcet chart, which implies the uniqueness of the situation at a given instant. The states are connected to each other by means of an evolution condition, which allows the passage from one situation to another one to be described.

For reasons of convenience, the behaviour description based on states is better replaced by a description based on steps. In the grafcet chart several steps may be active simultaneously, the situation being then characterized by the set of active steps at the considered moment. The evolution of one set of steps to another is translated by one or several transitions, each characterized by:

- its preceding steps,
- its succeeding steps,

- its associated transition-condition.

NOTE These reasons lead to the syntax rule enforcing the alternation step-transition.

4.3 GRAFCET, short presentation

4.3.1 General

The GRAFCET specification language is used for the design of grafcet charts to provide a graphical and synthetic representation of the sequential systems behaviour. The representation (see Figure 2) distinguishes:

- the structure, which allows possible evolutions between the situations to be described, and
- the interpretation, which enables the relationship between input, output variables and the structure (evolution, assignation and allocation rules are necessary to achieve this interpretation).

Symbols related to GRAFCET elements representing steps in a process and links between the steps are presented and exemplified in Tables 1 to 4 in Clause 5.

4.3.2 Structure

The structure comprises the following basic items:

- Step (definition: 3.1.8, symbol 1). A step is either active or inactive, the set of the active steps of a grafcet chart at any given instant represents the situation of this grafcet chart at this instant.
- Transition (definition: 3.1.10, symbol 7). A transition indicates that an evolution of the activity between two or more steps may evolve. This evolution is realized by the clearing of the transition.
- Directed link (definition: 3.1.2, symbol 10). A directed link connects one or several steps to a transition, or a transition to one or several steps.

4.3.3 Elements for interpretation

The following elements are used for the interpretation:

- Transition-condition (definition: 3.1.11, symbol 13). Associated with each transition, the transition-condition is a logical expression which is true or false and which is composed of input variables and/or internal variables.
- Action (definition: 3.1.1). The action indicates, in a rectangle, what shall be done to the output or internal variable, either by assignation (continuous action, symbol 20), or allocation (stored action, symbol 26).

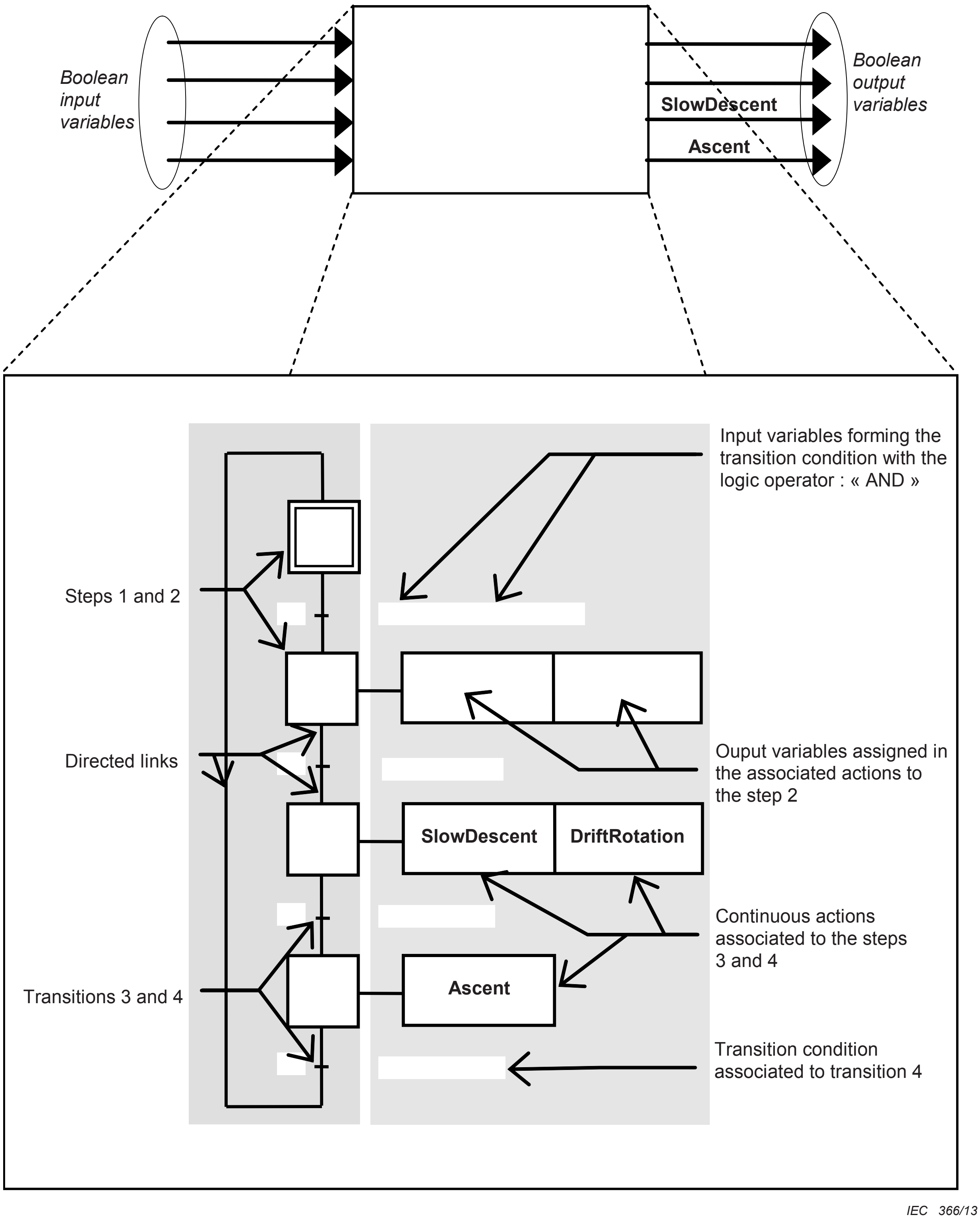


Figure 2 – Structure and interpretation elements used in a grafcet chart to describe the behaviour of a sequential part of the system defined by its input and output variables

4.4 Syntax rule

Step transition and transition step alternation shall always be respected whatever the sequence.

Consequences:

- Two steps shall never be connected directly by a directed link;
- The directed link shall only connect a step to a transition or a transition to a step.

4.5 Evolution rules

4.5.1 General

As each situation is characterized by the set of active steps at a given instant, the grafcet evolution rules only affect the application, on the steps, of the evolution principle between the situations of the sequential part of the system.

4.5.2 Initial situation

The initial situation is the situation at the initial time. Therefore, it is described by the set of steps active at this time. The choice of the situation at the initial time depends on the methodology relating to the type of sequential part of the system considered.

Rule 1: The initial situation, chosen by the designer, is the situation at the initial time.

4.5.3 Clearing of a transition

Rule 2: A transition is said to be enabled when all immediately preceding steps linked to this transition are active. The clearing of a transition occurs:

- when the transition is enabled, and
- when its associated transition-condition is true.

4.5.4 Evolution of active steps

Rule 3: The clearing of a transition simultaneously provokes the activation of all the immediate succeeding steps and the deactivation of all the immediate preceding steps.

4.5.5 Simultaneous evolutions

The evolution between two active situations implies that no other intermediate situation is possible and the change from one representation of the situation by a set of steps to another representation is instantaneous.

Rule 4: Several transitions, which can be cleared simultaneously, are simultaneously cleared.

4.5.6 Simultaneous activation and deactivation of a step

If a step is included in the description of the preceding situation and in that of the following one, it can therefore only remain active.

Rule 5: If during the operation, an active step is simultaneously activated and deactivated, it remains active.

4.6 Input events

4.6.1 General

The evolution rules show that only a change in the values of the input variables may cause the evolution of the grafcet chart. This change called "input event" shall be defined by the preceding value and the succeeding value of all the input variables to characterise this single event. In practice, a set of input events is specified only by the characterised state change (rising edge or falling edge) of one or several Boolean input variables.

NOTE The rising edge of a logical variable, indicated by the sign "↑" in front of a Boolean variable, indicates that this rising edge is only true for the change from value 0 to value 1 of the variable concerned. The falling edge of a logical variable noted by the sign "↓" in front of a Boolean variable, indicates that this falling edge is only true for the change from value 1 to value 0 of the variable concerned.

It is said that "the event occurs" at the date of the change of state of the input variables which characterize it.

4.6.2 Input events specification

The input events specification is implemented by a logical expression of one or several characteristic variables, often in a transition-condition. More rarely, it may also directly specify an internal event (see 4.7).

↑a EXAMPLE 1:

The expression "↑a" describes the set of all input events for which the preceding value of the input variable “a” is 0 and its succeeding value is 1, regardless of the value of the other input variables of the system.

a · ↑b EXAMPLE 2:

The expression "a × ↑b" describes the set of all input events for which the succeeding value of the input variable “a” is 1, and the preceding value of the input variable "b" is 0 and its succeeding value is 1, regardless of the value of the other input variables of the system.

a EXAMPLE 3:

The expression "a" describes the sets of all input events for which the succeeding value of the input variable “a” is 1, regardless of the value of the other input variables of the system.

NOTE Used in a transition-condition, this expression could lead to a transient evolution (see 3.12).

4.7 Internal events

4.7.1 General

Only certain input events could occur from a given situation. The connection between a situation and input event, which may occur from this situation, is called internal event (see 3.6). This notion is mainly used by the designer to condition an output allocation to a set of internal events (see 4.8.3). The description of a set of internal events is realized by one of the following ways.

4.7.2 Internal events described by the step activation

The step activation, noted graphically (symbol 27), describes the set of internal events each of which has this step activation as a consequence.

4.7.3 Internal events described by the deactivation of a step

The graphically noted deactivation of a step (symbol 28) describes the set of the internal events each of which have this step deactivation as consequence.

4.7.4 Internal events described by the clearing of a transition

The graphically noted clearing of a transition (symbol 29) describes the set of internal events each of which have the clearing of this transition as consequence.

4.8 Output modes

4.8.1 General

The actions enable links to establish the connection between the evolution of the grafcet chart and the outputs. Two output modes, continuous mode or stored mode, describe how the outputs depend on the evolution and on the system inputs.

4.8.2 Continuous mode (assignation on state)

In the continuous mode, the association of an action with a step indicates that an output variable has a true value if the step is active and if the assignation condition is verified. The assignation condition is a logical expression of the input variables and/or the internal ones (see symbol 22). If one of the conditions is not met and provided that no other action relating to the same output meets the conditions, the output variable concerned takes the false value.

Assignation refers to imposing the values of the output variables (true or false).

The set of the local assignation (relating to the active steps at a given instant) defines the assignation of all the output variables for this situation.

Assignation rule: for a given situation, the value of the outputs relating to the continuous actions is assigned:

- to the true value, for each output relating to the actions associated with active steps and for which the assignation conditions are verified,
- to the false value, for the other outputs (which are not assigned to the true value).

4.8.3 Stored mode (allocation on event)

In the stored mode, the association of an action to internal events is used to indicate that an output or internal variable takes and maintains the enforced value if one of these events occurs.

Explicit representations are necessary to describe the association of the actions with the events (activation step, deactivation step, clearing of a transition, etc.).

The value of an output or internal variable relating to a stored action remains unchanged until a new specified event modifies its value.

Allocation refers to storing, at a considered moment, a determined value affected to an output or internal variable.

Allocation rule: the value of an output or internal variable, relating to a stored action and associated to an event, is allocated to the indicated variable, if the specified internal event occurs; the value of this variable is false (Boolean variable) or null (numeric variable) at the initialisation.

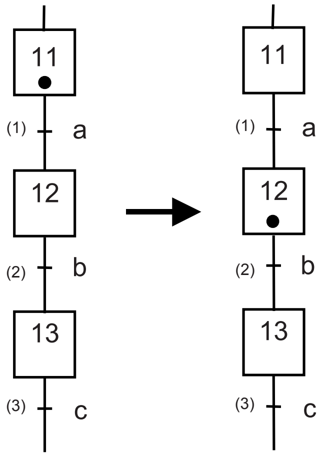
4.9 Application of the evolution rules

4.9.1 General

Intuitive interpretation of the evolution, called “step by step”, designates the progressive way which allows, on the occurrence of an input event and from the preceding situation, to determine the succeeding situation of this event, by the successive application of the evolution rules on each transition. The interpretation facility is a device to enable an indirect specification of the evolution, but the designer shall take care that the clearing of the transitions on this path does not involve the effective activation of the intermediate situations.

4.9.2 Non transient evolution

In general, the evolution is non-transient, which means that the input event only leads to one evolution stage (the simultaneous clearing of one or more transitions).



EXAMPLE: "Non transient evolution"

Preceding situation: step 11 active, a = 0, b = 0 and c = 0.

Intuitive interpretation of the evolution:

The change in the value “a” involves the clearing of the transition (1) and the activation of the step 12, the transition (2) cannot be cleared, because b = 0, the subsequent situation is therefore: step 12 active.

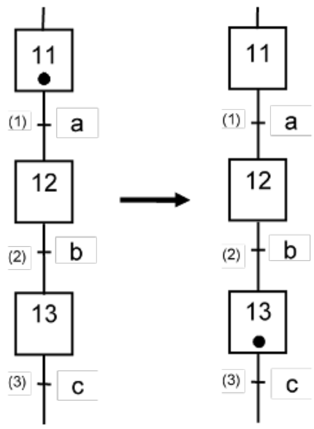
Real interpretation of the evolution:

The occurrence of one of the input events such as the value of “a” changes from 0 to 1 leads straight to the subsequent situation: step 12 active.

4.9.3 Transient evolution

In some cases, the application of the evolution rules can lead to successively clearing some transitions (in several evolution stages) if the transition-conditions associated with the subsequent transitions are already true, when the first transitions considered are cleared. The corresponding description, referred to as transient, uses the path taken to indicate how to move from a preceding situation to a succeeding situation (see 3.9).

The corresponding intermediate steps, referred to as unstable are not activated, but we consider that they have been "virtually" activated and deactivated along the intuitive evolution path, as well as for the corresponding transitions which have been "virtually" cleared.



EXAMPLE: "Transient evolution"

Preceding situation: step 11 active, a = 0, b = 1 and c = 0.

Intuitive interpretation of the evolution:

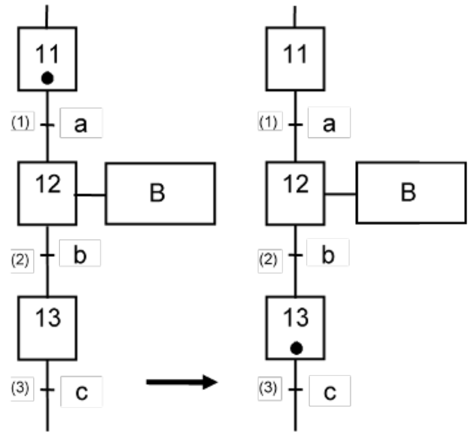
The change in the value “a” involves the clearing of the transition (1) and the virtual activation of the step 12, then the transition (2) is virtually cleared, because b=1, leading to the succeeding situation: step 13 active.

Real interpretation of the evolution:

The occurrence of one of the input events, such as the value of "a" changes from 0 to 1, leads to the succeeding situation: step 13 active.

4.9.4 Consequence of a transient evolution on the assignations

The assignation of an output value by a continuous action associated with a step, which is an unstable step in the case of a transient evolution, is not effective, since the step is not really activated (see 4.8.2).



EXAMPLE: "Continuous action associated with an unstable step"

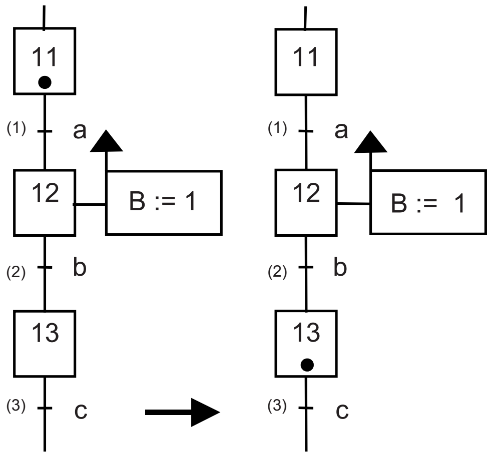
Preceding situation: step 11 active, a = 0, b = 1 and c = 0.

The occurrence of one of the input events such as the value of "a" changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The preceding situation (step 11 active) and the succeeding situation (step 13 active) assign the value 0 to the output variable B. The unstable step 12 being not really activated, the assignation of B to the value 1 is not effective on the transient evolution.

4.9.5 Consequence of a transient evolution on the allocations

The allocation to a determinate value of an output by a stored action (symbol 26) associated to a step, which is an unstable step in the case of a transient evolution, is effective since this allocation is associated with the events releasing this evolution (see 4.8.3).

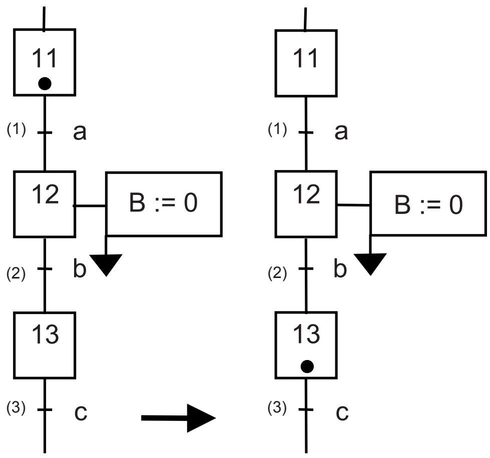


EXAMPLE 1: "Stored action associated with the activation of an unstable step"

Preceding situation: step 11 active, a = 0, b = 1 and c = 0.

The occurrence of one of the input events such as the value of "a" changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The allocation of the value 1 to the output variable B is realized on the occurrence of one of the input events having the real or the virtual activation of the step 12 as consequence.



EXAMPLE 2: "Stored action associated with the deactivation of an unstable step"

Preceding situation: step 11 active, a = 0, b = 1 and c = 0.

The occurrence of one of the input events such as the value of "a" changes from 0 to 1, leads straight to the subsequent situation: step 13 active.

The allocation of the value 0 to the output variable B is realized on the occurrence of one of the input events having the real or the virtual deactivation of the step 12 as consequence.

4.10 Comparison between the two output modes

4.10.1 General

The choice of the output mode depends on the practice and methodology used. However, the designers' attention is drawn to the important differences between the two modes.

4.10.2 Determination of the value of the outputs

Depending on the chosen mode the determination of the value of the outputs are described as:

- in continuous mode, all the outputs are assigned according to the situation, to the true value for the outputs explicitly indicated in the actions associated to the active steps, and to the false value for the other ones which are implicitly set by omission (see assignation rule, 4.8.2);
- in the stored mode, only the considered outputs are modified according to the indicated value, the other stored values of the outputs remain unchanged (see allocation rule, 4.8.3).

4.10.3 Analysis of the value of the outputs for a grafcet chart at a defined instant

Depending on the chosen mode the analysis of the value of the outputs are described as:

- in the continuous mode, the knowledge of the situation and the value of the inputs is sufficient to determine the value of the outputs (see 4.8.2);
- in the stored mode, the knowledge of the situation and the value of the inputs is not sufficient, the preceding evolutions shall also be known to determine the value of the outputs (see 4.8.3).

4.10.4 Actions relative to transient evolution

Depending on the chosen mode the actions relative to transient evaluation are described as:

- in the continuous mode, the actions associated with an unstable step are not taken into consideration because this step is not activated (see 4.9.2);
- in the stored mode, the actions associated with events and in relation with a transient evolution are taken into consideration because the triggered events releasing this evolution occur (see 4.9.3).

4.10.5 Possible conflict on the value of the outputs

Depending on the chosen mode the possible conflict on the value of the outputs are managed as:

- in the continuous mode, the assignation principles ensure every assignation conflict on the particular output to be avoided;
- in the stored mode, the allocation rules do not allow the possible assignation conflicts on a same output to be avoided. The designer shall ensure that two contradictory allocations cannot occur simultaneously.

NOTE 1 Both output modes can be used in one specification in GRAFCET, but the value of an output variable is determined either by assignation or by allocation. The specification of an allocation to an output variable (stored mode), excludes this output variable of any assignation (continuous mode).

NOTE 2 Clause 5 gives the graphic symbols which enable the stored actions (indicated by explicit representation according to the set of specified events) to be distinguished from the continuous ones (indicated by absence of any representation).

NOTE 3 In the frequent case of the specification of control system behaviour, the current industrial practice forces the employment of the continuous mode for all the Boolean outputs to the actuators, and the stored mode for describing internal control tasks. These tasks, such as the incrementation of a counter, or the modification of the value for a numerical register, refer to internal variables, which are not necessarily Boolean ones. The internal tasks associated with the stored actions, as well as the calculation of expressions associated with transition-conditions, are not described in the present standard, but are associated by the use of the logical description of the grafcet evolutions.

5 Graphical representation of the elements

The elements of GRAFCET have their own symbolic representation which, when correctly associated, enable clear and synthetic function-charts to be implemented.

NOTE 1 Only the global representation of the symbols is imposed; dimensions and details (thickness of lines, font of characters, etc.) are left up to the users.

NOTE 2 The stippled representation indicates the context of the symbol.

Table 1 – Steps

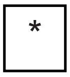
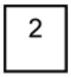
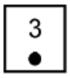
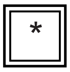
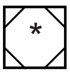
No.	Symbol	Description
[1]		**Step:** At a given moment, a step is either active or inactive. The set of active steps defines the situation of the given system at the considered instant. The height-width ratio of the rectangle is arbitrary, although a square is recommended. For the purposes of identification, the steps shall have a label, for example, alphanumerical. The label assigned to the step shall replace the asterisk at the upper half of the general symbol. EXAMPLE 1: "Step 2"“  EXAMPLE 2: "Step 3 represented in its active state"  NOTE It could be useful to indicate which steps are active at a given instant by marking these steps with a dot. This dot is not part of the step symbol and is only used for explanatory purposes.
[2.1]	X*	**Step variable:** The active or inactive state of the step may be represented by the logical values "1" or "0" respectively of a Boolean variable X*, in which the asterisk * shall be replaced by the label of the relevant step. EXAMPLE: “Step variable of the step 8” X8
[2.2]	T*	**Step duration**: The duration of an active step may be represented by the value of a time variable T*, in which the asterisk * shall be replaced by the label of the relevant step. EXAMPLE: “Step duration of step fill“ TFill.
[3]		**Initial step:** This symbol means that this step participates in the initial situation. NOTE 1 The rules of symbol 1 apply. NOTE 2 An initial step could be “unstable”, see 4.9.3. EXAMPLE: “Initial step 12“ 
[4]		**Enclosing step:** This symbol indicates that this step contains other steps referred to as enclosed steps. NOTE 1 The rules of symbol 1 apply. NOTE 2 The properties and the examples of the use of the enclosing step are given in 7.4.
[5]		**Initial enclosing step:** This symbol means that this enclosing step participates in the initial situation. NOTE An initial enclosing step contains at least one enclosed initial step.
[6]		**Macro-step:** Unique representation of a detailed part of the function-chart referred to as the expansion of the macro-step. NOTE The properties and the examples of the use of the macro-step are given in 7.5.

Table 2 – Transitions

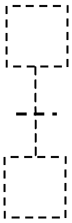
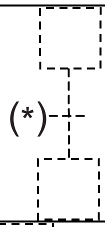
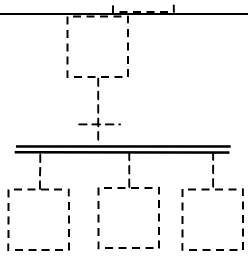
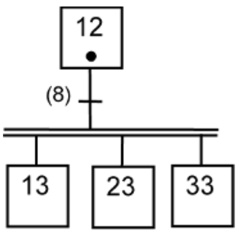
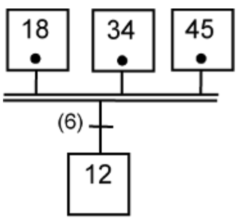
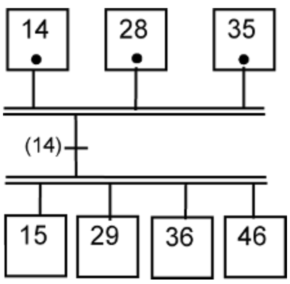
No.	Symbol	Description
[7]		**Transition** from one step to another: A transition is represented by a line perpendicular to the link joining two steps. NOTE 1 The transition is enabled when the immediate preceding step is active (see the evolution rule No. 2, 4.5.3). NOTE 2 *Only one transition* is ever possible between two steps (see 4.4). NOTE 3 It is possible, for graphical representation reasons, to place transitions on horizontal directed links (see Figure B5, partial grafcet G1). NOTE 4 The symbolism of transitions is not subject of this standard. Transitions can be described by plain text, Boolean expressions, logic charts, etc.
[8]		**Transition designation:** The transition may have a designation, generally placed to the left, which should not be mistaken for the associated transition-condition. An alphanumerical designation for the transition shall replace the asterisk.
[9]		**Synchronization** preceding and/or succeeding a **transition:** When several steps are connected to the same transition, the directed links from and/or to these steps are grouped, to succeed or precede the synchronization symbol represented by two parallel horizontal lines. NOTE The reference for the synchronization symbol is 9.2.2.5 of ISO 5807:1985.  EXAMPLE 1: Transition from one step (12) to several (13, 23, 33). The transition (8) is enabled when the step 12 is active.  EXAMPLE 2: Transition from several steps (18, 34, 45) to one step (12). The transition (6) is only enabled when all preceding steps are active.  EXAMPLE 3: Transition from several steps (14, 28, 35) to several steps (15, 29, 36, 46). The transition (14) is only enabled when all preceding steps are active.

Table 3 – Directed links

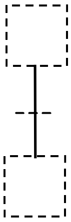
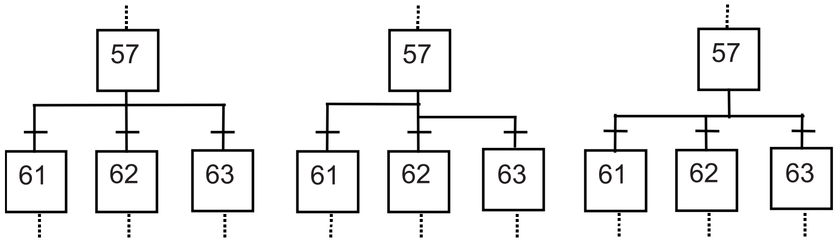
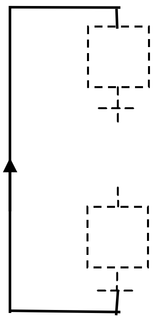
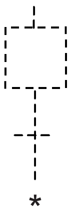
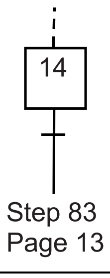
No.	Symbol	Description
[10]		**Directed link from top to bottom:** The evolution paths between the steps are indicated by directed links connecting steps to transitions and transitions to steps. Directed links are horizontal or vertical. Diagonal links are only permitted in those rare cases where they improve the clarity of the chart. Crossovers of vertical and horizontal links are permitted if no relationship exists between those links. Accordingly such crossovers shall be avoided when the links are related to the same evolution. EXAMPLE: The three representations are permitted but the representations 2 and 3 are recommended to avoid misunderstanding between links with and without relationship.  (1) (2) (3)
[11]		**Directed link from bottom to top**: By convention, the direction of the evolution is always from top to bottom. Arrows shall be used if this convention is not respected or if their presence enables a clearer understanding.
[12]		**Linked label:** If a directed link has to be broken (for example in complex charts or when one chart covers several pages) the number of the destination steps and the number of the page on which it appears, shall be indicated. The asterisk shall be replaced by the linked label. EXAMPLE: Evolution to step 83 of page 13. 

Table 4 – Associated transition-conditions

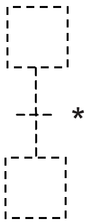
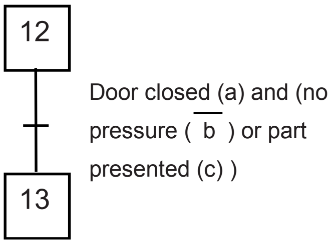
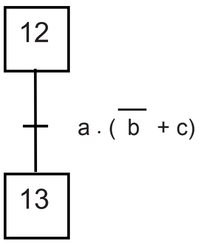
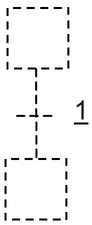
No.	Symbol	Description
[13]		**Transition-condition:** A logical proposition, called a transition-condition, which can be either true or false, is associated with each transition. If a corresponding logical variable exists, it is equal to 1 when the transition-condition is true or equal to 0 when the transition-condition is false. The logical proposition forming the transition-condition comprises one or several Boolean variables, (input variable, step variable, predicate value, etc.). The asterisk shall be replaced by the description of the transition-condition in the form of text, of a Boolean expression, or by using graphical symbols.  EXAMPLE 1: Transition-condition described by a text.  EXAMPLE 2: Transition-condition described by a Boolean expression.
[14]		**Transition-condition always true:** The notation "1" means that the transition-condition is always true. NOTE In this case, the evolution is to be transient (see 4.9.3), the clearing of the transition is only conditioned by the activity of the preceding step.

Table 4 *(continued)*

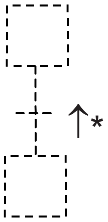
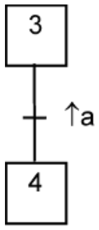
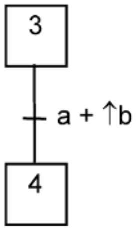
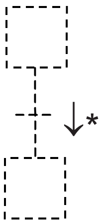
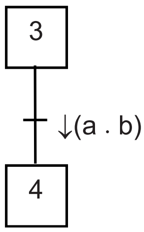
No.	Symbol	Description
[15]		**Rising edge of a logical variable:** The notation " ↑ " means that the transition-condition is only true at the change of the state of the variable * (rising edge: changing from value 0 to value 1, see the note in 4.6). This symbol is general and applies to all logical propositions, either for an elementary variable or for a set of several Boolean variables.  EXAMPLE 1: The associated transition-condition is only true when a changes from state 0 to state 1. NOTE By applying the evolution rule No. 2, the transition is only cleared on a rising edge of a after the transition has been enabled by the activity of step 3.  EXAMPLE 2: The associated transition-condition is true only when a is true or when b changes from state 0 to state 1.
[16]		**Falling edge of a logical variable:** The notation " ↓ " means that the transition-condition is only true on the change of the state of the variable * (falling edge: changing from value 1 to value 0, see the note in 4.6). This symbol is general and applies to all logical propositions, either for an elementary variable or for a set of several Boolean variables.  EXAMPLE: The associated transition-condition is true only when the logical product "a · b" changes from state 1 to state 0.

Table 4 *(continued)*

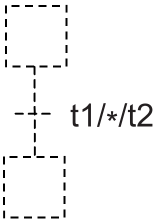
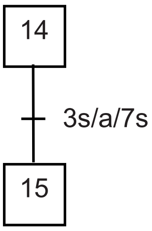
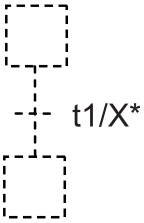
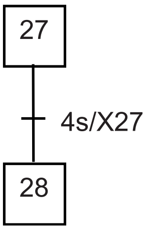
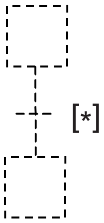
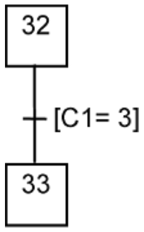
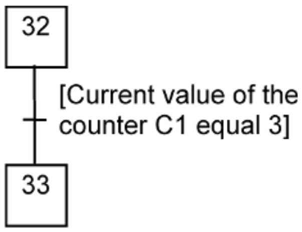
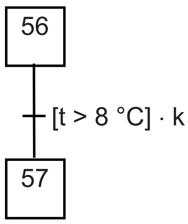
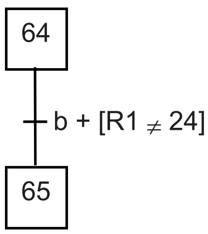
No.	Symbol	Description
[17]		**Time dependent transition-condition:** The notation " t1/*/t2 " indicates that the transition-condition is true only after a time t1 from the occurrence of the rising edge (↑*) of the time limited variable and becomes false again after a time t2 from the occurrence of the falling edge (↓*). The asterisk shall be replaced by the time-delayed variable, for example a step variable or an input variable. t1 and t2 shall be replaced by their real value expressed in the selected time unit. The time-delayed variable shall remain true for a period equal to or greater than t1 for the transition-condition be true. NOTE This notation is that of the delay element defined by the standard IEC 60617-S01655 (2004-09).  EXAMPLE: The transition-condition is true 3 s after the change of "a" from state 0 to the state 1, it becomes false 7 s after the change of “a” from state 1 to the state 0.
[18]		**Usual simplification of symbol 17:** Current use is to delay the step variable by a time t2 equal to zero, then, the transition-condition becomes false on deactivation of the step * that activated the delay. The asterisk shall be replaced by the label of the step which is required to be delayed. The time delayed step shall remain active during a time equal to or greater than t1 for the transition-condition to be true. This notation can be used when the time-delayed step is not the preceding step of the transition.  EXAMPLE: The transition-condition will be true during 4 s after the activation of step 27, and will be false with the clearing of the transition which deactivates the preceding step. In this case, the duration of the activity of the step 27 is 4 s.

Table 4 *(continued)*

No.	Symbol	Description
[19]		**Boolean value of a predicate:** "[*]" indicates that the Boolean value of the predicate constitutes the transition-condition variable. Therefore, when the assertion * is verified, the predicate has value of 1, otherwise the predicate has a value of 0. The asterisk shall be replaced by the assertion, which shall be tested. The Boolean variable of the predicate can be associated with other logical variables to constitute a logical proposition of transition-condition.  EXAMPLE 1: The transition-condition is true when the assertion "C1=3" is verified.  EXAMPLE 1a: The transition-condition is true when the current value of the counter C1 is equal to 3. NOTE The form of the assertion is not imposed; for example a literal language can be used.  EXAMPLE 2: The transition-condition is true when the assertion " t > 8 °C " is verified and when the Boolean variable k has a value of "1", that means, when the temperature t is higher than the value 8 °C and when the high level k is reached.  EXAMPLE 3: The transition-condition is true when the Boolean variable "b" has a value of 1 or when the assertion "R1 ≠ 24" is verified, that means when the part is at the place b, or when the register R1 has not yet reached the value of 24.

Symbols representing action are presented and exemplified in Table 5 and Table 6 below. Actions can be of type continuous actions (see Table 5) or stored actions (see Table 6).

A stored action has a label (symbol 26) situated in the rectangle which describes how the output variable is allocated to a determinate value according to the allocation rule (see 4.8.3).

The event specification associated with the stored action is necessary to indicate when the corresponding output allocations occur (see allocation rule 4.8.3). Four means of description (symbols 27 to 29) allow the easy specification of different sets of internal events associated with the stored actions.

Table 5 – Continuous actions

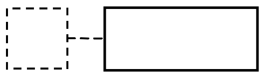
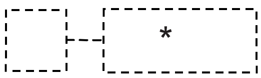
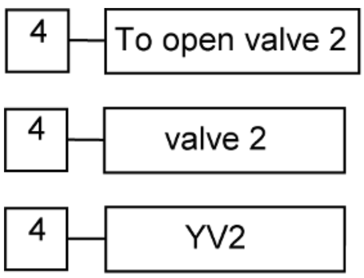
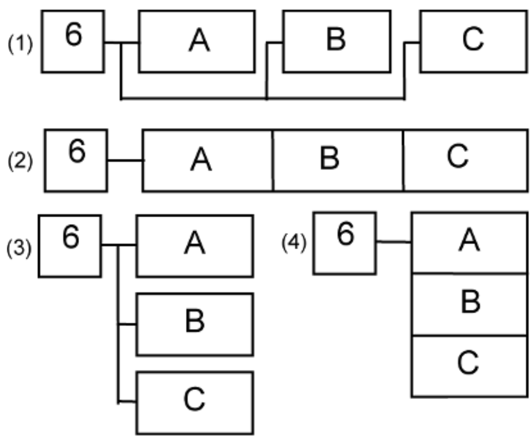
No.	Symbol	Description
[20]		**Continuous action:** A continuous action is necessarily associated with a step. Several actions can be associated with one step. The height-width ratio is arbitrary although a rectangle of the same height as the step is recommended. In the absence of an explicit symbolisation of a stored action (symbols 27 to 29), the general rectangular symbol associated with a step always designates a continuous action.
[21]		**Assignation label of an output:** Each action shall have a label inside the rectangle, which refers to this action. The label of a continuous action is the designation of the output variable assigned to the true value according to the assignation rule (see 4.8.2). The asterisk shall be replaced by the wording of the output variable. The textual expression of the label can take an imperative or indicated form, the only important point is the reference to the output. The order in which the actions are represented does not imply any sequence between the actions. EXAMPLE 1: Different forms, literal or symbolic, of an action label which refers to the output when the value is true, will provoke valve 2 to open.  EXAMPLE 2: Different representations (1, 2, 3, 4) of the association of several actions at one step.  NOTE The four representations are strictly equivalent. Representation (2) and (4) can be considered respectively as simplifications of the representation (1) and (3).

Table 5 *(continued)*

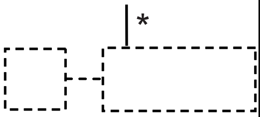
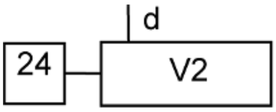
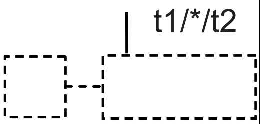
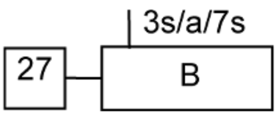
No.	Symbol	Description
[22]		**Assignation condition:** A logical proposition, called an assignation condition, which can be true or false, influences any continuous action. The absence of notation indicates that the condition is always true. The assignation condition description in text format or a Boolean expression between the input variables and/or the internal variables shall replace the asterisk. This assignation condition shall never include an edge of variable (see symbols 15 and 16), because the continuous action is of course not memorised, an assignation on event having no meaning (see 4.8.3). EXAMPLE 1: Output V2 is assigned to the true value when step 24 is active and when the assignation condition d is true. In the opposite case, output V2 is assigned to the false value. In other words (as a Boolean equation): V2 = X24 · d  NOTE X24 is the step variable which reflects the activity of step 24. EXAMPLE 2: Output V2 is assigned to the true value when step 24 is active (the assignation condition is always true). In the opposite case, output V2 is assigned to the false value. In other words (as a Boolean equation): V2 = X24 
[23]		**Time dependent assignation condition:** The notation "t1/*/t2" indicates that the assignation condition is true only after a time t1 from the occurrence of the rising edge (↑*, see symbol 15) of the timed variable * and becomes false again after a time t2 from the occurrence of the falling edge (↓*, see symbol 16) The asterisk shall be replaced by the timed variable, for instance a step variable or an input variable. t1 and t2 shall be replaced by their real value expressed in the selected time unit. The limited variable shall remain true for a time equal to or greater than t1 for the assignation condition to be true. NOTE This notation is that of the delay element defined by IEC 60617-S01655 (2004-09). EXAMPLE: The assignation condition is true only 3 s after "a" changes from state "0" to state "1", and false 7 s after "a" changes from state "1" to state "0".  The value of output B depends on the activity of step 27 and on the value of the assignation condition (sees assignation rules 4.8.2).

Table 5 *(continued)*

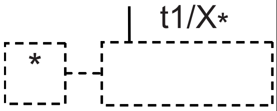
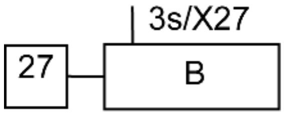
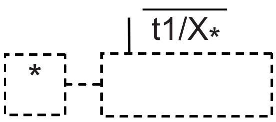
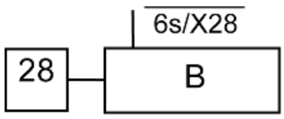
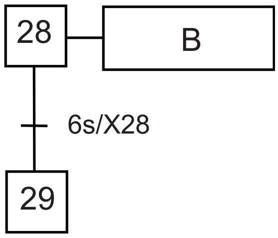
No.	Symbol	Description
[24]		**Delayed action:** The delayed action is a continuous action in which the assignation condition is true only after a time t1 specified from the activation of the associated step * , with the objective of delaying the assignation to the true value of the corresponding output. EXAMPLE: Output B is assigned to the true value when 3 s have elapsed since the activation of step 27.  NOTE If the step 27 activity time is less than 3 s, then the output B variable is not assigned to the true value.
[25]		**Time limited action:** The time limited action is a continuous action in which the assignation condition is true for a period of time t1 specified from the activation of the associated step *, for limiting the duration of the assignation to the true value of the corresponding output**.** EXAMPLE 1: Output B is only assigned to the true value for 6 s from the activation of step 28.  NOTE If the step 28 activity time is less than 6 s, the output B variable is assigned to the true value only during the step 28 activity time. **Equivalent representation:** The simplified delay operator can be used in the associated transition-condition for the succeeding step to limit the allocation time of the true value to the corresponding output (see symbol 18). EXAMPLE 2: Equivalent representation of the example 1 with the symbol 18. Output B is only assigned to the true value for 6 s from the activation of step 28. 

Table 6 – Stored actions

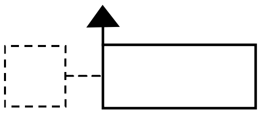
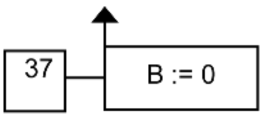
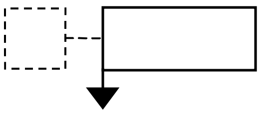
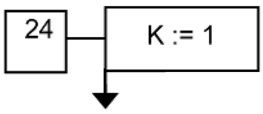
No.	Symbol	Description
[26]	* := #	**Allocation of the value # to a variable *:** The wording indicates, for a stored action, the setting to the value **#** of a variable * when one of the events associated with the action occurs (see allocation rule 4.8.3). The stored action supporting this allocation shall be associated with the internal events specification (symbols 27 to 29). The allocation can be described textually within the action rectangle. A := 1 EXAMPLE 1: Set the value of a Boolean variable A to true. The wording " A:= 1 " describes the allocation of the value 1 to a Boolean variable A when one of the events associated with the action occurs. b := 0 EXAMPLE 2: Set the value of a Boolean variable b to false. The wording " b:= 0 " describes the allocation of the value 0 to a Boolean variable b when one of the events associated with the action occurs. C := C+1 EXAMPLE 3: Incrementation of a counter The wording "C:= C+1" describes the allocation of the value C+1 to a numeric variable C when one of the events associated with the action occurs.
[27]		**Action on activation:** An action on activation is a stored action associated with the set of the internal events, which have, for each one, the linked step activation as consequence. The traditional representation of the action by a rectangle is completed, on the left side, by an arrow symbolising the activation of the step.  EXAMPLE: The Boolean variable B is allocated to the value 0 when one of the events, leading to the activation of step 37, occurs.
[28]		**Action on deactivation:** An action on deactivation is a stored action associated with the set of the internal events, which have, for each one, the linked step deactivation as consequence. The traditional representation of the action by a rectangle is completed, on the left side, by an arrow symbolizing the deactivation of the step.  EXAMPLE: The Boolean variable K is allocated to the value 1 when one of the events, represented by the deactivation of step 24, occurs.

Table 6 *(continued)*

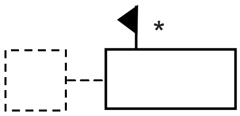
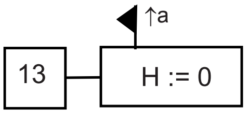
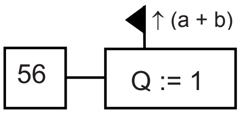
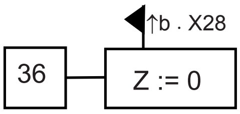
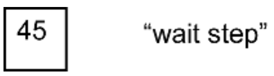
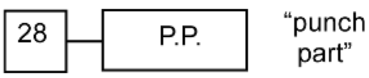
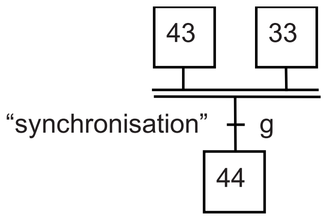
No.	Symbol	Description
[29]		**Action on event:** An action on event is a stored action associated with each of the internal events described by the expression * on condition that the step, with which the action is connected, is active. The traditional representation of the action by a rectangle is completed, on top, by a symbol indicating that the action is conditioned by the occurrence of one of the internal events specified by the expression *. It is recommended that the logical expression *, which shall describe a set of internal events, is made up of one or more input variable edges.  EXAMPLE 1: The Boolean variable H is allocated to the value 0 when one of the events, represented by " ↑a ", occurs and simultaneously, step 13 is active. NOTE The combination between the set of the input events, represented by the expression " ↑a ", and the step 13 activity represents in fact a set of internal events (see definition 3.6).  EXAMPLE 2: The Boolean variable Q is allocated to the value 1 when one of the events, represented by " ↑(a + b)", occurs and simultaneously, step 56 is active.  EXAMPLE 3: The Boolean variable Z is allocated to the value 0 when one of the events, represented by " ↑b ", occurs and simultaneously, steps 36 and 28 are active.

Table 7 contains comments associated with GRAFCET elements.

Table 7 – Comments associated with elements of a grafcet chart

No.	Symbol	Description
[30]	"*"	**Comment:** A comment concerning the graphic elements of a function-chart shall be placed between inverted commas (quotation mark). The asterisk shall be replaced by the comment.  EXAMPLE 1: Comment "wait step" referring to step 45.  EXAMPLE 2: Comment "punch part" referring to the action associated with step 28.  EXAMPLE 3: Comment "synchronisation" referring to a transition.

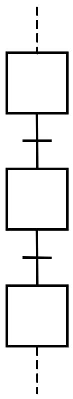
6 Graphical representation of sequential structures

6.1 General

The designer can construct grafcet charts using different distinctive structures, subject to strict application of the syntax rule concerning step/transition alternation.

6.2 Basic structures

6.2.1 Sequence



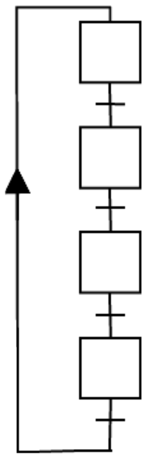
A sequence is a succession of steps such that:

- each step, except the last one, has only one succeeding transition,
- each step, except the first one, has only one preceding transition enabled by a single step of the sequence.

NOTE 1 The sequence is said to be "active" if at least one of its steps is active. The sequence is said to be "inactive" when none of its steps is active.

NOTE 2 A sequence may include any number of steps.

6.2.2 Cycle of a single sequence



The case of a looped sequence such that:

- each step has only one succeeding transition,
- each step has only one preceding transition enabled by a single step of the sequence.

NOTE 1 A cycle of a single sequence may constitute a partial grafcet (see 7.2.2).

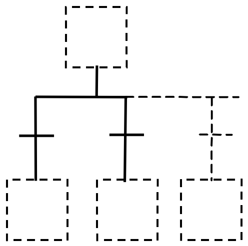
NOTE 2 A cycle of a single sequence shall satisfy at least one of the following conditions to allow the activation of its steps:

– to have at least one initial step,

– to be submitted by a forcing order from a partial grafcet at a higher level (see 7.3),

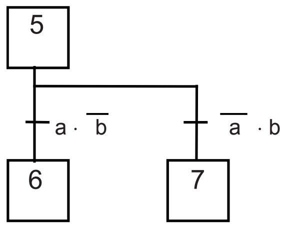
– to belong to one of the encapsulations of an enclosing step (see 7.4).

6.2.3 Selection of sequences

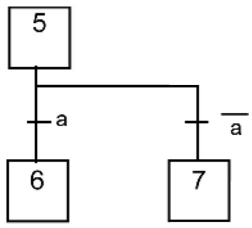


The selection of sequences shows a choice of evolution between several sequences starting from one or several steps. This structure is represented by as many simultaneously enabled transitions as possible evolutions.

NOTE Exclusive activation of a selected sequence is not guaranteed from the structure. The designer should ensure that the timing, logical or mechanical aspects of the transition-conditions are mutually exclusive.

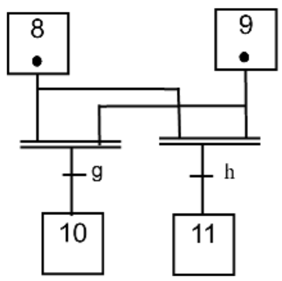


EXAMPLE 1: The exclusion between the sequences is achieved by the logical exclusion of the two receptivities. If “a” and “b” are simultaneously true when step 5 is active, no transition may be cleared.



EXAMPLE 2: Priority sequence.

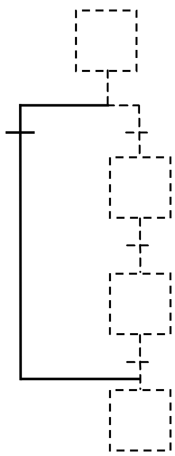
In this example, a priority is given to the transition 5/6, which is cleared when “a” is true.



EXAMPLE 3: Selection of sequences following synchronization of two preceding sequences.

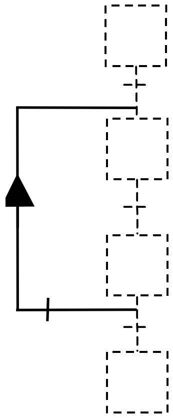
The selection of the succeeding sequences, by g and h, is possible only when the two transitions are cleared by the simultaneous activity of steps 8 and 9 (symbol 9).

6.2.4 Step skip



Particular case of selection of sequences, which allows a complete sequence or one or several steps of the sequence to be skipped, when, for example, the actions associated to these steps become unnecessary.

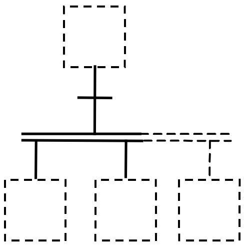
6.2.5 Backward sequence skip



Particular case of selection of sequences, which enables a sequence to be repeated until, for example, an established condition is satisfied.

NOTE It is possible, for graphical representation reasons, to place transitions on horizontal directed links (see Note 3 symbol 7).

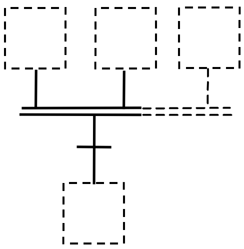
6.2.6 Activation of parallel sequences



The synchronisation symbol 9 is used in this structure to indicate the simultaneous activity of several sequences from one or several steps.

NOTE After their simultaneous activation, the evolution of the active steps in each of the parallel sequences thus becomes independent.

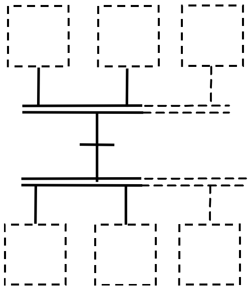
6.2.7 Synchronization of sequences



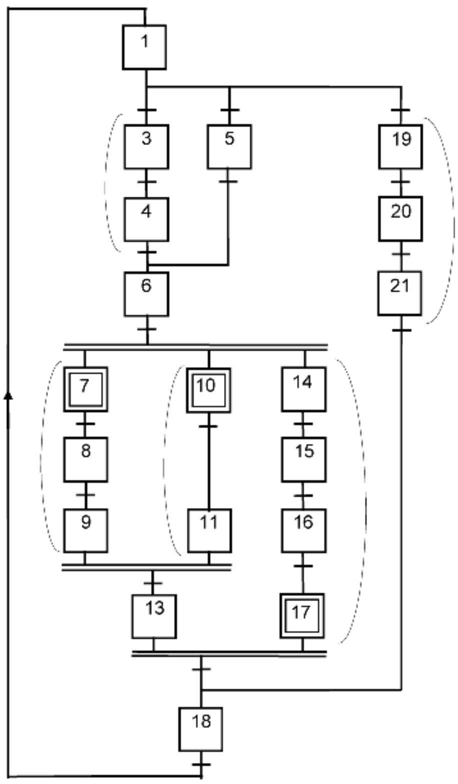
The synchronisation symbol 9 is used in this structure to indicate the delay before preceding sequences end before the activation of the succeeding sequence.

NOTE The transition is only enabled when all the preceding steps are active.

6.2.8 Synchronization and activation of parallel sequences



The synchronisation symbol 9 is used twice in this structure to indicate the delay before preceding sequences end before the simultaneous activation of the succeeding sequences.



EXAMPLE: Grafcet in which the following basic structures can be distinguished:

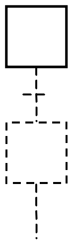
- the sequences (some of them are marked by parentheses),
- a selection of sequences (from step 1 to steps 3, 5 and 19),
- an activation of the parallel sequences (downstream from step 6),
- two synchronisations of sequences (from steps 9 and 11 to step 13, and from steps 13 and 17 to step 18).

NOTE 1 This example shows only the structure of the grafcet, its interpretation is not described.

NOTE 2 This grafcet is not a typical example because a grafcet is not necessarily looped back.

6.3 Particular structures

6.3.1 Starting of a sequence by a source step



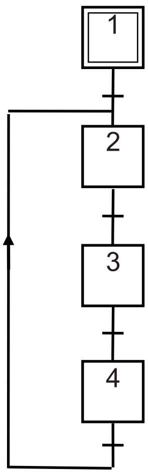
A source step is a step which does not have any preceding transition.

NOTE 1 To allow the activation of the source step, at least one of the following conditions, shall be satisfied:

the source step is initial,

the source step is required by a forcing order from a partial grafcet of the higher level (see 7.3),

the source step is one of the activated steps of an enclosure (see 7.4).

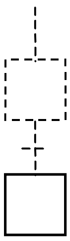


EXAMPLE 1: Initial source step:

The initial source step 1 is only active at the initialization time, the steps 2, 3, and 4 form a cycle of a single sequence.

NOTE 2 Only the grafcet structure is represented, its interpretation is not described.

6.3.2 End of a sequence by a pit step



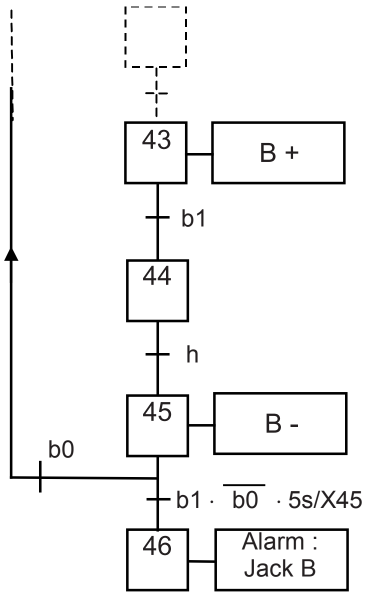
A pit step is a step which does not have any succeeding transition.

NOTE 1 The deactivation of the pit step is possible by only one of the two following ways:

a forcing order from a higher level partial grafcet (see 7.3),

the deactivation of the enclosing step if the pit step is enclosed there (see 7.4).

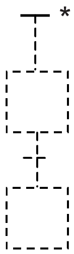
NOTE 2 A step may be source and pit at the same time, it then forms a single step sequence used to show a combinatorial behaviour.



EXAMPLE: Pit step:

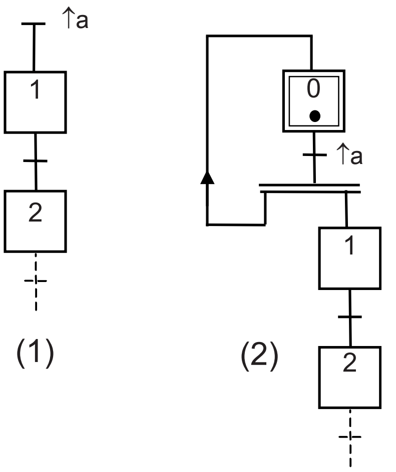
Pit step 46 is only activated if the logical condition "b1 · $\overline{b0}$ " is verified 5 s after the activation of step 45 (see symbol 18). The output "Alarm: Jack B" is then assigned the true value.

6.3.3 Starting of a sequence with a source transition



A source transition is a transition, which does not have any preceding step. By convention, the source transition is always enabled and it is cleared as soon as its transition-condition * is true.

NOTE 1 The activation of the succeeding step of a source transition is effective as long as its transition-condition remains true, independent of the state of the transition-conditions for transitions enabled by this step (see evolution rule No. 5, 4.5.5). To avoid a continuous activation of the succeeding step of the source transition, it is better for the associated transition-condition to become true only when an input event or an internal event occurs. For that, the logical expression forming the transition-condition shall always include an input edge.

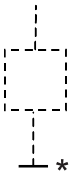


EXAMPLE: Source transition and equivalent structure:

Representations (1) and (2) describe an equivalent behaviour: step 1 is activated each time the Boolean variable a changes from value 0 to value 1. The representation (1) uses the source step, the representation (2) uses the synchronization symbol and a loop back to maintain initial step 0 active.

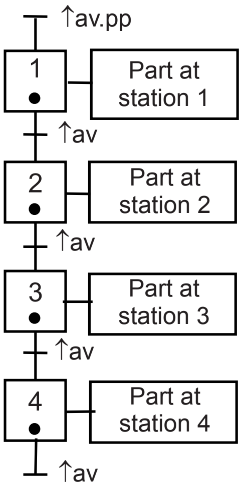
NOTE 2 The dot in step 0 indicates that this step is permanently active.

6.3.4 End of a sequence by a pit transition



A pit transition is a transition, which has no succeeding step.

NOTE 1 When the pit transition is enabled and when its associated transition-condition * is true, the only consequence of the clearing of the transition is the deactivation of the upstream steps.



EXAMPLE: structure of a shift register:

The structure of a shift register is a pertinent use of a source transition and of a pit transition. In this example, each active step indicates the presence of a part at the corresponding station. The presence of a part (pp) at the entry and the advance of the transfer between stations (↑av) activates step 1 by the clearing of the source transition. On each advance of the transfer (↑av) the enabled transitions are simultaneously cleared, including the pit transition downstream of the step 4.

NOTE 2 The representation corresponds to the frequent case when all the steps are simultaneously active.

7 Structuring

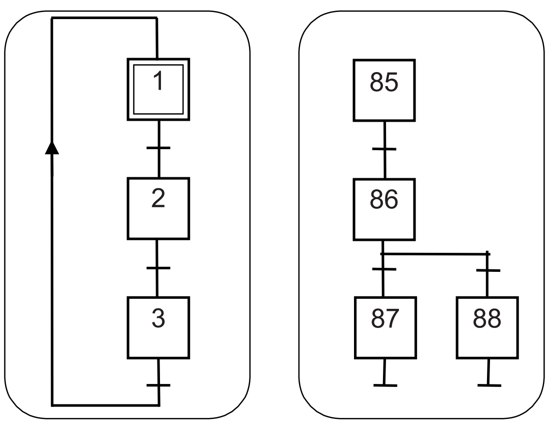
7.1 General

The complexity of the automated systems requires means for the structuring of the specification. This structuring assisted or not by suitable methodologies, can be limited simply to the division of the specification or can integrate hierarchical concepts of forcing or enclosure.

7.2 Partition of a grafcet chart

7.2.1 Connected grafcet chart

A connected grafcet chart is a structure in which there is always a continuity of links (alternation of steps and transitions) between any two elements, step or transition, in the grafcet chart.



(1) (2)

EXAMPLE:

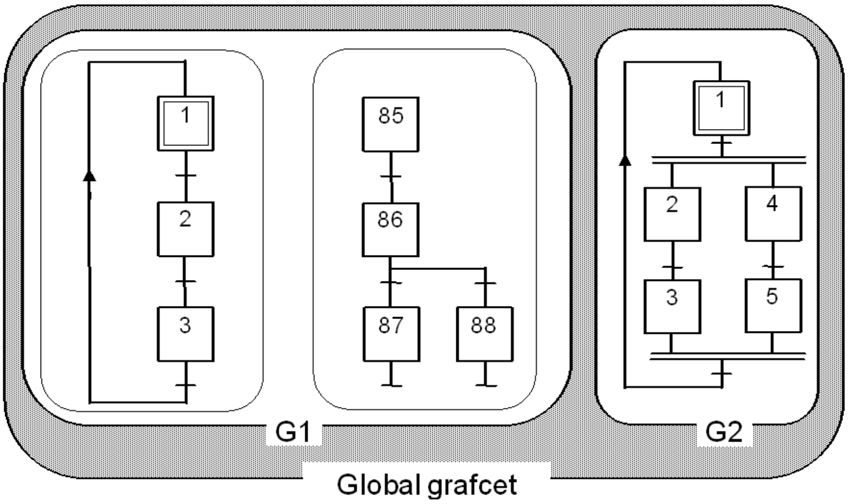
All the elements in frame (1) form a connected grafcet chart, since its steps and its transitions are connected by a directed link. The elements of the frame (2) also form a connected grafcet chart.

NOTE A non-connected grafcet has no technological meaning.

7.2.2 Partial grafcet

Formed by one or several connected grafcet charts, a partial grafcet chart (see Table 8) will results from the division of the global grafcet chart describing the behaviour of the sequential part of a system, according to the methodological criteria.

A partial grafcet chart is placed in a frame in which the label G* of the represented grafcet chart is noted on the lower border.



EXAMPLE: Partition of a global grafcet:

The global grafcet is made of the partial grafcets G1 and G2. The partial grafcet G1 is made of two connected grafcets.

Table 8 – Partial grafcet chart

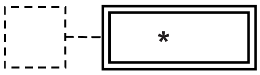
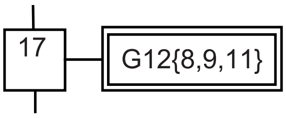
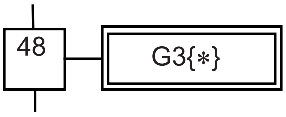
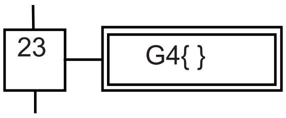
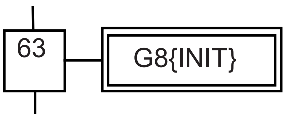
No.	Symbol	Description
[31]	G*	Name of a partial grafcet: By convention, the letter G indicates a partial grafcet. The asterisk shall be replaced by the name of the partial grafcet.
[32]	XG*	Partial grafcet variable: A partial grafcet is said to be active when at least one of its steps is active, consequently it is said to be inactive when none of its steps is active. The active or inactive state of a partial grafcet may be respectively represented by the logical values "1" or "0" of a variable XG* in which the asterisk * is replaced by the name of the considered partial grafcet. The asterisk shall be replaced by the name of the partial grafcet. EXAMPLE: XG1 indicates the variable of the partial grafcet 1.
[33]	G#{…,…}	Situation of a partial grafcet: The situation of a partial grafcet is represented by the set of its active steps at the considered moment. The situation of a partial grafcet # is given by G#{…,…} the contents of the curly brackets lists the active steps characterizing the situation of the partial grafcet at the considered moment. The sign # shall be replaced by the name of the partial grafcet. EXAMPLE: G12{8, 9, 11} indicates the situation of the partial grafcet 12 at the considered moment, the situation in which only the steps 8, 9 and 11 are active.
[34]	G#{*}	Current situation of a partial grafcet: The asterisk indicates, by default, the situation of the partial grafcet # at the considered moment. The sign # shall be replaced by the name of the partial grafcet.
[35]	G#{ }	Empty situation of a partial grafcet: Designates the situation of the partial grafcet # when none of its steps is active. The sign # shall be replaced by the name of the partial grafcet.
[36]	G#{INIT}	Initial situation of a partial grafcet: Designates the situation of a partial grafcet # at the initial moment. The sign # shall be replaced by the name of the partial grafcet.

7.3 Structuring using the forcing of a partial grafcet chart

This means of structuring the specification of the sequential part of a system uses forcing orders (see Table 9). These orders allow the imposition of a specific situation to a given partial grafcet chart, from the situation of another (see example, Annex B). A partial grafcet chart may be forced by different preceding steps and orders.

NOTE "Orders are similar to continuous actions. While an action influences a variable, an order influences a partial grafcet chart."

Table 9 – Forcing of a partial grafcet chart

No.	Symbol	Description
[37]		Forcing order of a partial grafcet: Symbol in which the asterisk shall be replaced by a situation of a partial grafcet (symbols 33 to 36). Associated with the activity of a step of a hierarchically higher partial grafcet, the forcing order is an internal order which allows the imposition of a situation on a hierarchically lower partial grafcet. The forcing order is represented in a double rectangle associated with the step, to distinguish it from an action. The forcing order is an internal order for which the execution has priority on the application of the evolution rules. The forced grafcet cannot evolve during the period of the forcing order. The grafcet is said to be frozen. The use of forcing orders in a specification requires a hierarchical structure using partial grafcets in such a way that every forcing partial grafcet is at a higher level than the partial forced grafcets.  EXAMPLE 1: Forcing of a partial grafcet to an explicit situation. When step 17 is active, the partial grafcet 12 is forced to the situation characterised by the activity of the steps 8, 9 and 11.  EXAMPLE 2: Forcing of a partial grafcet to the current situation. When step 48 is active, the partial grafcet 3 is forced to the situation in which it is at the forcing time. NOTE 1 This order is also named “freezing order”.  EXAMPLE 3: Forcing of a partial grafcet to the empty situation. When step 23 is active, the partial grafcet 4 is forced to the empty situation (see symbol 35). NOTE 2 In this case, no step of G4 is active.  EXAMPLE 4: Forcing of a partial grafcet to the initial situation. When step 63 is active, the partial grafcet 8 is forced to the situation in which only its initial steps are active.

7.4 Structuring using the enclosure

A set of steps (partial grafcet) are said to be enclosed by a step, referred to as the enclosing step (see Table 10), if and only if, when this enclosing step is active, at least one of the enclosed steps is active. The designer can use the enclosure to structure a grafcet chart hierarchically (see example in Figure 3 or Figure B6). An enclosure can only belong to one enclosing step.

Table 10 – Enclosing steps

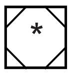
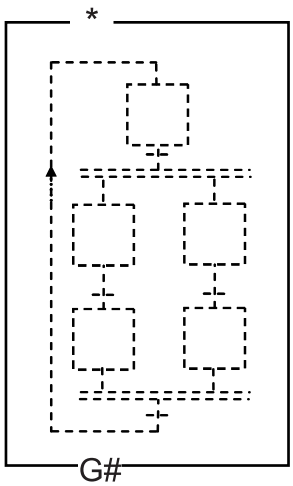
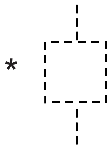
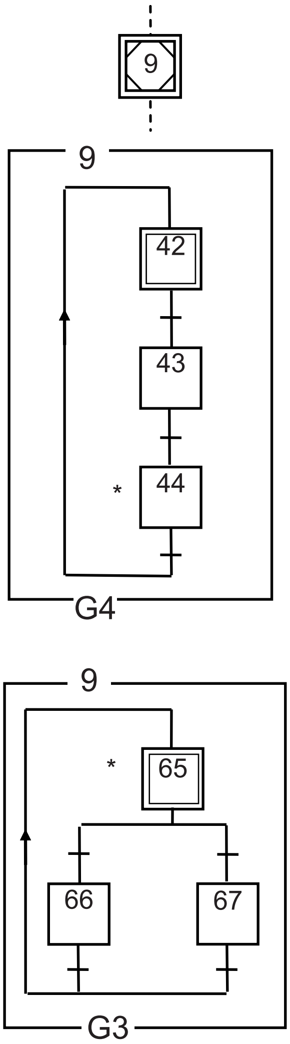
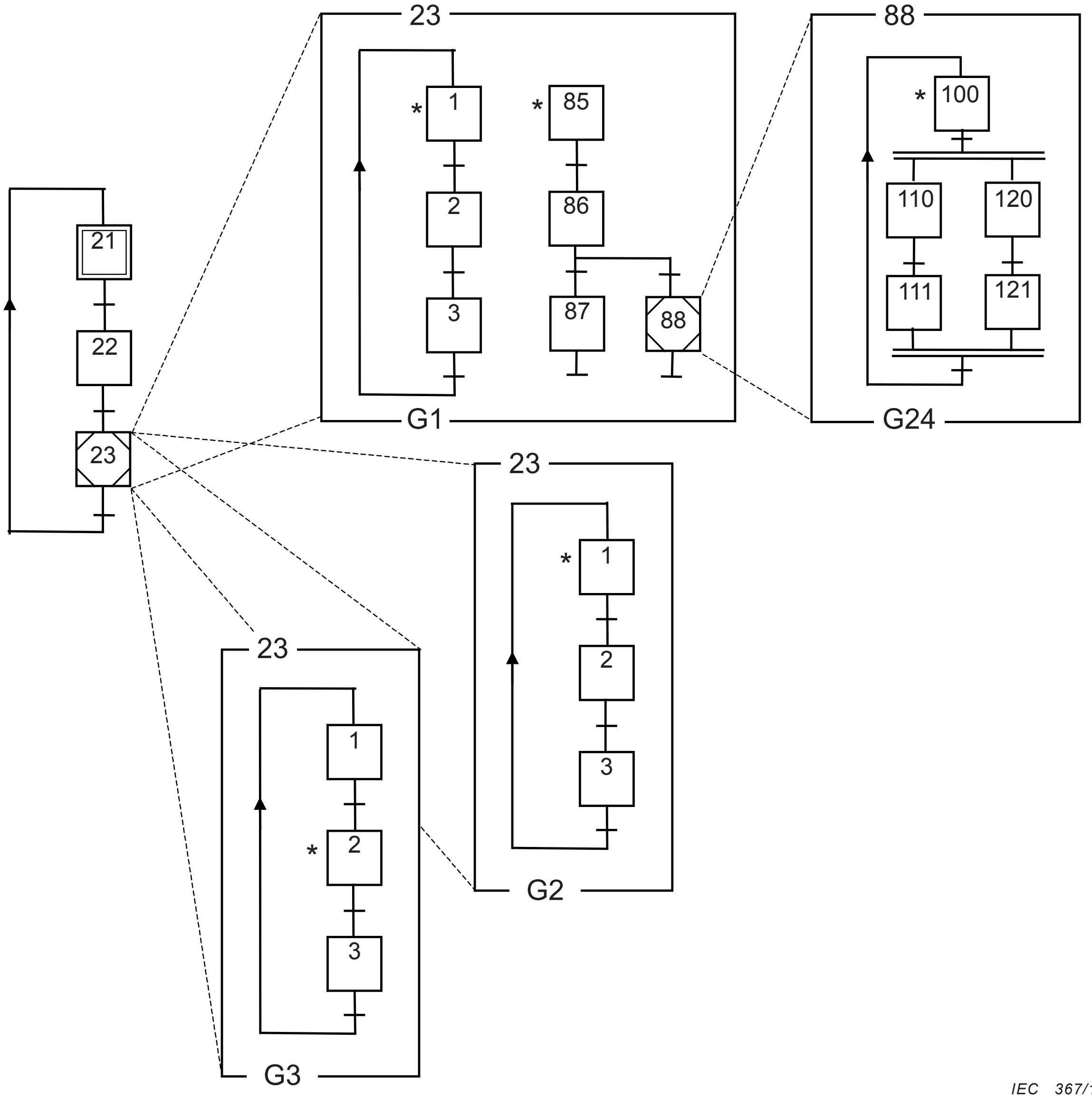
No.	Symbol	Description
[4]		Enclosing step (reminder of symbol 4): This notation indicates that this step contains other steps, referred to as enclosed, in one or several enclosures of the same enclosing step. The enclosing step has all the properties of the step, the step label shall replace the asterisk. The enclosing step may lead to one or several enclosures having at least one active step when the enclosing step is active in each of them, and having no active step when the enclosing step is inactive.
[38]		Graphic representation of an enclosure: An enclosure **#** of an enclosing step * may be represented by the partial grafcet of the enclosed steps grouped in a frame in which the name * of the enclosing step is placed at the top left side and the label G**#** of the represented enclosure is placed on the lower left side. In an enclosure, the set of the enclosed steps shall constitute a partial grafcet chart whose name may be used as a label for the corresponding enclosure.
[39]	X*/G#	Global designation of an enclosure: An enclosure **#** of an enclosing step * may be globally described by a textual expression in which the enclosing step ***** is designated by the step variable X*, the enclosure by the symbol /, and the enclosed steps by the name of the partial grafcet G# to which they belong. NOTE This representation supposes that the designated partial grafcet was previously defined.
[40]	X*/X#	Elementary designation of an enclosure: A textual expression can indicate that a step # is enclosed in an enclosing step ***** by using the step variables, without naming the enclosure. NOTE This notation suits the design of a hierarchical series of steps enclosed in one another. It also allows a relative identification of the steps by level of enclosure. EXAMPLE: X4/X25/X12 designates the enclosure of step 12 in step 25, which is itself enclosed in step 4.

Table 10 *(continued)*

No.	Symbol	Description
[5]		Initial enclosing step (reminder of symbol 5): This representation indicates that this step participates in the initial situation. In that case, at least one of the steps enclosed in each of its enclosures shall also be an initial step.
[41]		Activation link, general symbol. Represented by an asterisk at the left of the symbols of the enclosed steps, the activation link indicates the enclosed steps which are active at the activation of the enclosing step. The activation link should not be confused with the symbol for the initial steps, which may be enclosed. However, it is possible that an enclosed initial step also has an activation link. The deactivation of an enclosing step leads to the deactivation of all its enclosed steps. This deactivation often occurs at the clearing of the succeeding transition of the enclosing step but may also result from any other means of deactivation (forcing or enclosing of a higher level).  EXAMPLE: The enclosing step 9 is necessarily an initial step, because it encloses the initial step 42. The enclosure G4, of the enclosing step 9, contains steps 42, 43 and 44. The initial step 42 participates in the initial situation, it is therefore active at the initial time. On the other hand and for every activation of the step 9, and following of the grafcet evolution, step 44 is activated. The enclosure G3, of the enclosing step 9, contains steps 65, 66 and 67. The initial step 65 participates in the initial situation, it is therefore active at the initial time. It is also activated for every activation of step 9, as a consequence of the grafcet evolution.



EXAMPLE: Structuring by enclosure:

The enclosing step 23 has three enclosures represented by the partial grafcet G1,G2 and G3. The partial grafcet 24 is enclosed in step 88 of the partial grafcet 1. When the enclosing step 23 is activated, the steps 1 and 85 of G1 are also activated (the same is true for the other enclosing steps of 23: G2 and G3). Thus the partial grafcets G1, G2 and G3 are executed in parallel.

When the enclosing step 88 is activated, step 100 of G24 is also activated.

The deactivation of step 88 deactivates all steps of G24.

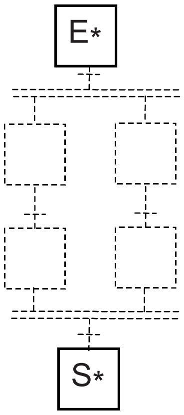
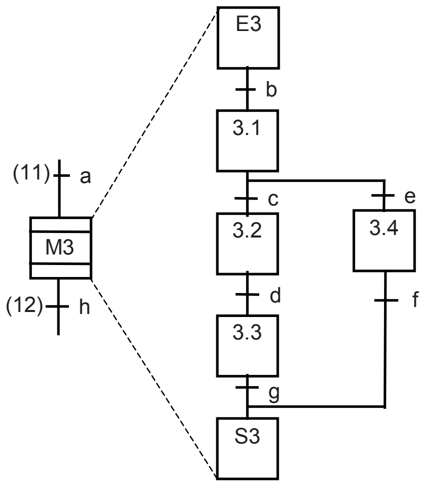
The deactivation of step 23 deactivates all the steps of G1, G2, G3 and, if step 88 was active, all the steps of G24.

Figure 3 – Example of grafcet with enclosures (including description)

7.5 Structuring using the macro-steps

To improve the understanding of grafcet charts, the specifications can be represented on several levels by "macro-representation" depicting the function to be performed without worrying about all the details, which are superfluous at the actual state of the description. The use of macro-steps (see Table 11) allows a gradual description from the general to the particular.

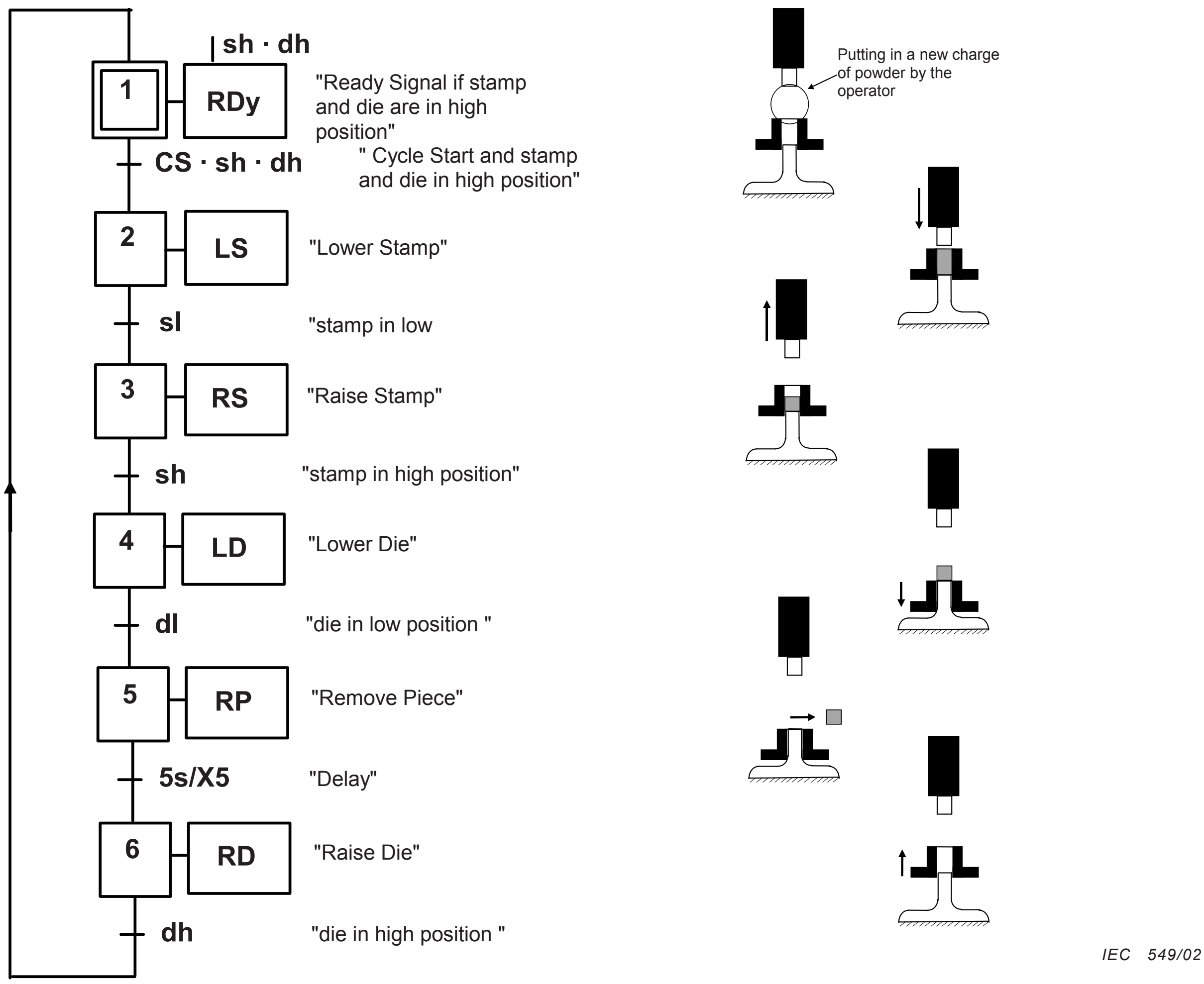
Table 11 – Macro-steps

No.	Symbol	Description
[6]		Macro-step (reminder of symbol 6): Unique representation of a detailed part of the grafcet referred to as the expansion of the macro-step. An expansion can only belong to just one macro step. The macro-step does not have all the properties of the other kinds of step (symbols 1 to 5) because only its exit step (see symbol 42) can validate its succeeding transitions. A macro-step label shall replace the asterisk.
[42]		Expansion of the macro-step: The expansion of a macro-step M* is a part of the grafcet with one entry step E* and one exit step S*. The entry step E* becomes active when one of the preceding transitions of the macro-step is cleared. One or all-succeeding transitions of the macro-step can be enabled only when the exit step S* is active. NOTE 1 The expansion of a macro-step can consist of one or several initial steps. NOTE 2 The expansion of a macro-step can consist of one or several macro-steps.  EXAMPLE: Macro-step M3 represented with its expansion: The clearing of the transition 11 leads to the activity of the entry step E3 of the macro-step M3. Transition 12 is enabled only when exit step S3 is active. The clearing of transition 12 leads to the inactivity of step S3.
[43]	XM*	Macro-step variable: A macro-step is said to be active when, at least, one of its steps is active, consequently it is said to be inactive when none of its steps is active. The active or inactive state of a macro-step can be represented by the logical values "1" or "0" respectively of a variable XM* in which the asterisk * shall be replaced by the name of the considered macro-step.

Annex A
(informative)

Example of the control of a press

A press for compressed powders works as shown on the grafcet chart of Figure A.1. When the press is in stand-by at step 1, the stamp and the die are in high position and a "ready" signal is lighted to indicate to the operator to put in a new charge. The actions are successively performed as shown on the grafcet chart.



Codes:

Inputs		**Outputs**	
CS	Cycle Start	RDy	Ready signal
sh	stamp in high position	LS	Lower Stamp
sl	stamp in low position	RS	Raise Stamp
dh	die in high position	LD	Lower Die
dl	die in low position	RP	Remove Piece
		RD	Raise Die

Figure A.1 – Representation of the working press using a grafcet

Annex B
(informative)

Example: Automatic weighing-mixing

B.1 Presentation of the example

Products A and B, previously weighed on a weighing unit C, and soluble bricks, brought one by one on a belt, are fed into a mixer N. The automatic system described in Figure B.1 allows a mixture of these three components to be obtained.

B.2 Cycle

Actuating the push-button "cycle start" causes the simultaneous weighing of products and the transport of bricks as follows:

- weighing product A up to the mark "a" of the weighing unit, and then dosing product B up to the mark "b" followed by emptying weighing unit C into the mixer;
- transport of two bricks.

The cycle ends with the mixer rotation and its final tipping after time "*t*1". The rotation of the mixer continues during emptying.

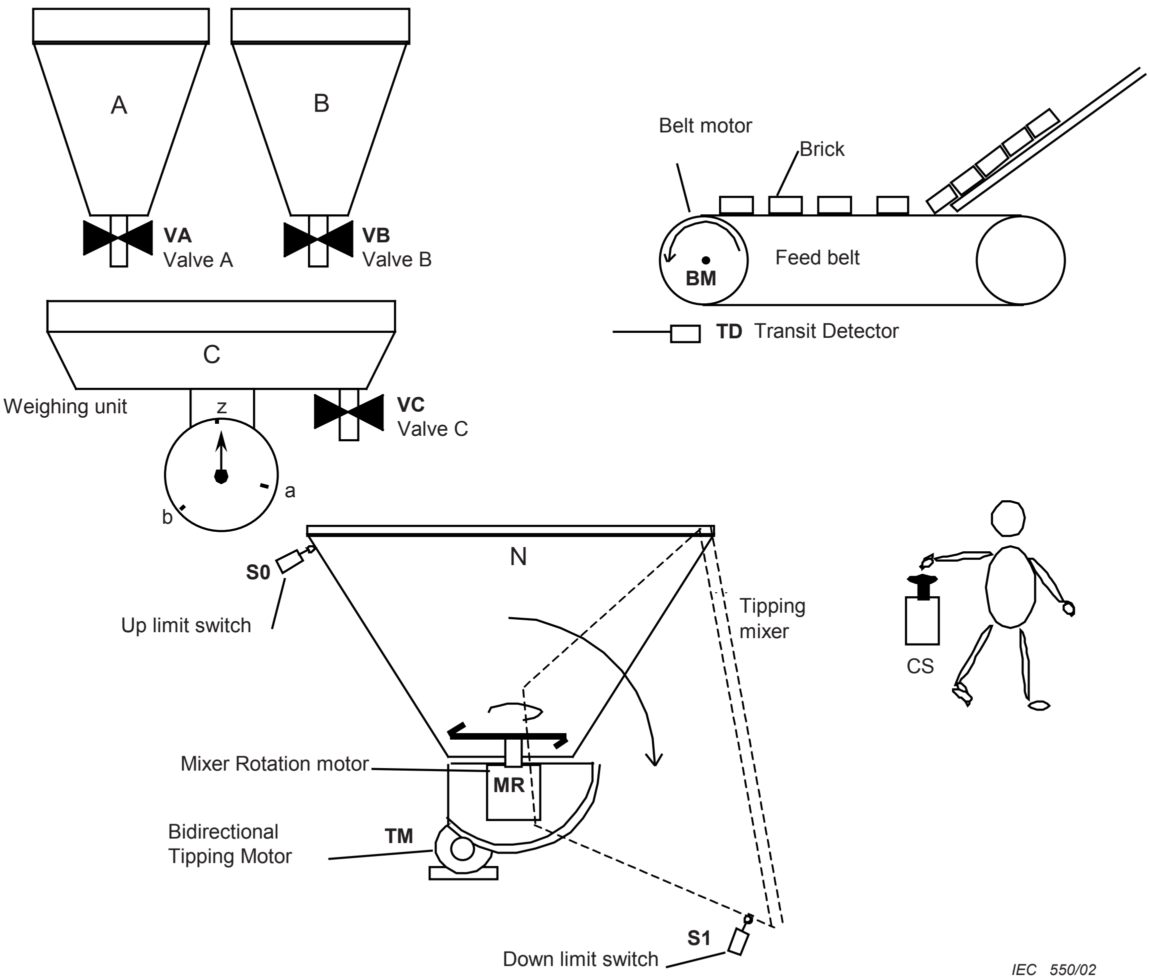
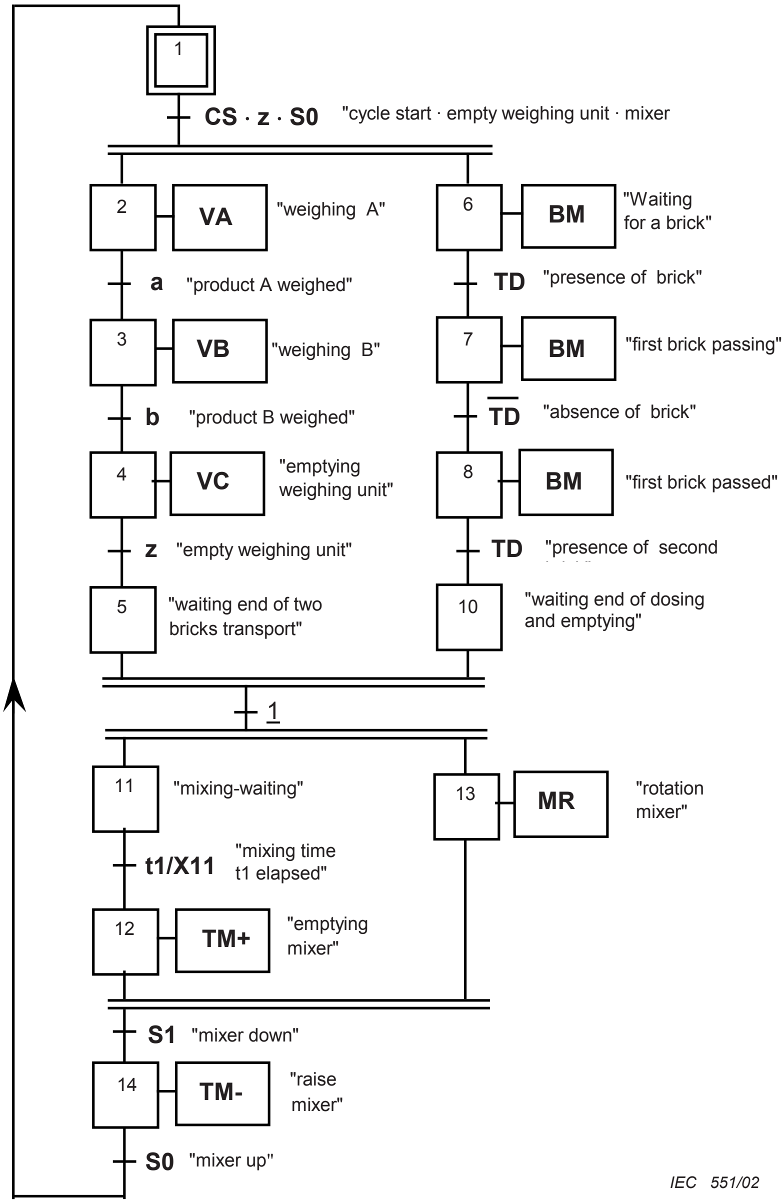


Figure B.1 – Overview diagram of weighing-mixing system

B.3 Behaviour description of the control command of the weighing-mixing

The logical behaviour of the weighing-mixing control command can be described by any of the grafcet charts of Figures B.2 to B.4.



Codes:

Inputs / **Outputs**

CS	Cycle Start	BM	Belt Motor
TD	Transit Detector	MR	Mixer Rotation motor
a	Fluid weight A reached	TM+	Tipping Motor (down)
b	Fluid weight A + B reached	TM-	Tipping Motor (up)
z	Empty weighing unit	VA	Opening Valve A
S0	Mixer up	VB	Opening Valve B
S1	Mixer down	VC	Opening Valve C

Figure B.2 – Grafcet of a weighing-mixing involving only continuous actions

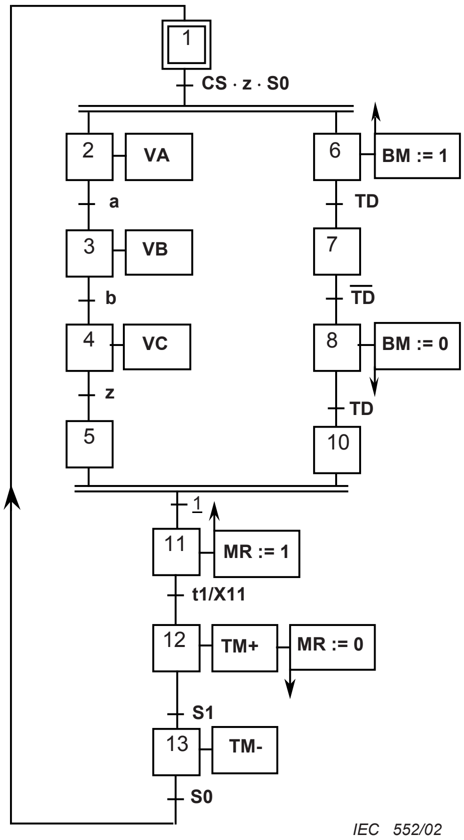


Figure B.3 – Grafcet of the weighing-mixing, involving continuous and stored actions

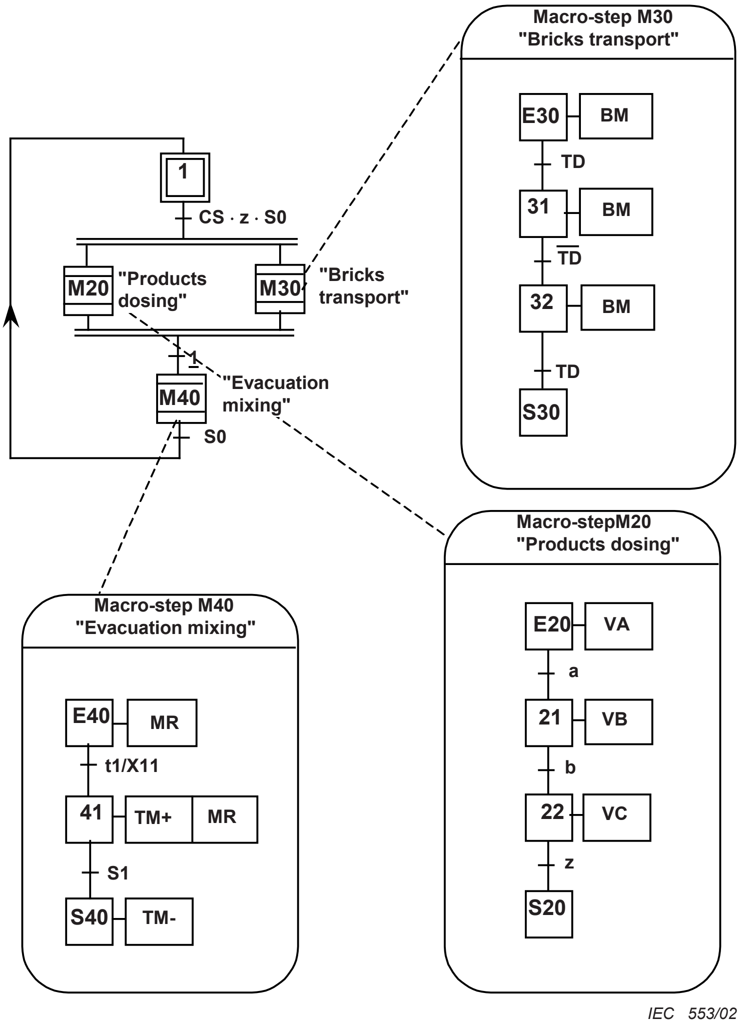
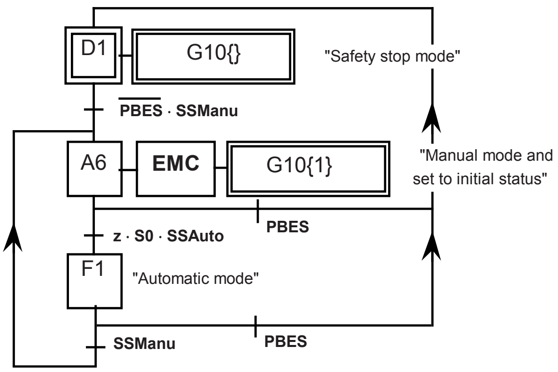


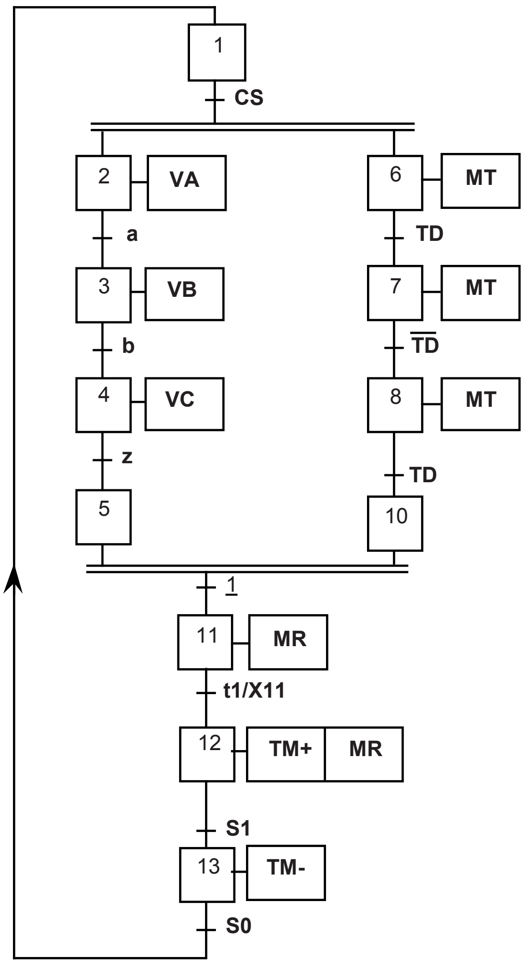
Figure B.4 – Grafcet of the weighing-mixing, divided into a global description using macro-steps and a description detailed by the macro-step expansions

B.4 Weighing-mixing: structuring according to operating modes

Taking into account the operating modes of the automatic weighing-mixing can lead to the hierarchical structuring of the specification by using forcing orders (Figure B.5) or enclosing steps (Figure B.6). The complementary inputs and outputs given in the table in Figure B.5 are necessary to take into account the orders from the operator.



G1 : Partial grafcet of operating (upper hierarchical level)



G10 : Partial grafcet of automatic cycle (lower hierarchical level)

IEC 554/02

Codes:

Inputs		**Outputs**	
PBES	Push-button emergency stop	EMC	Enabled manual controls
SSAuto	Selector-switch on auto mode		
SSManu	Selector-switch on manual mode		

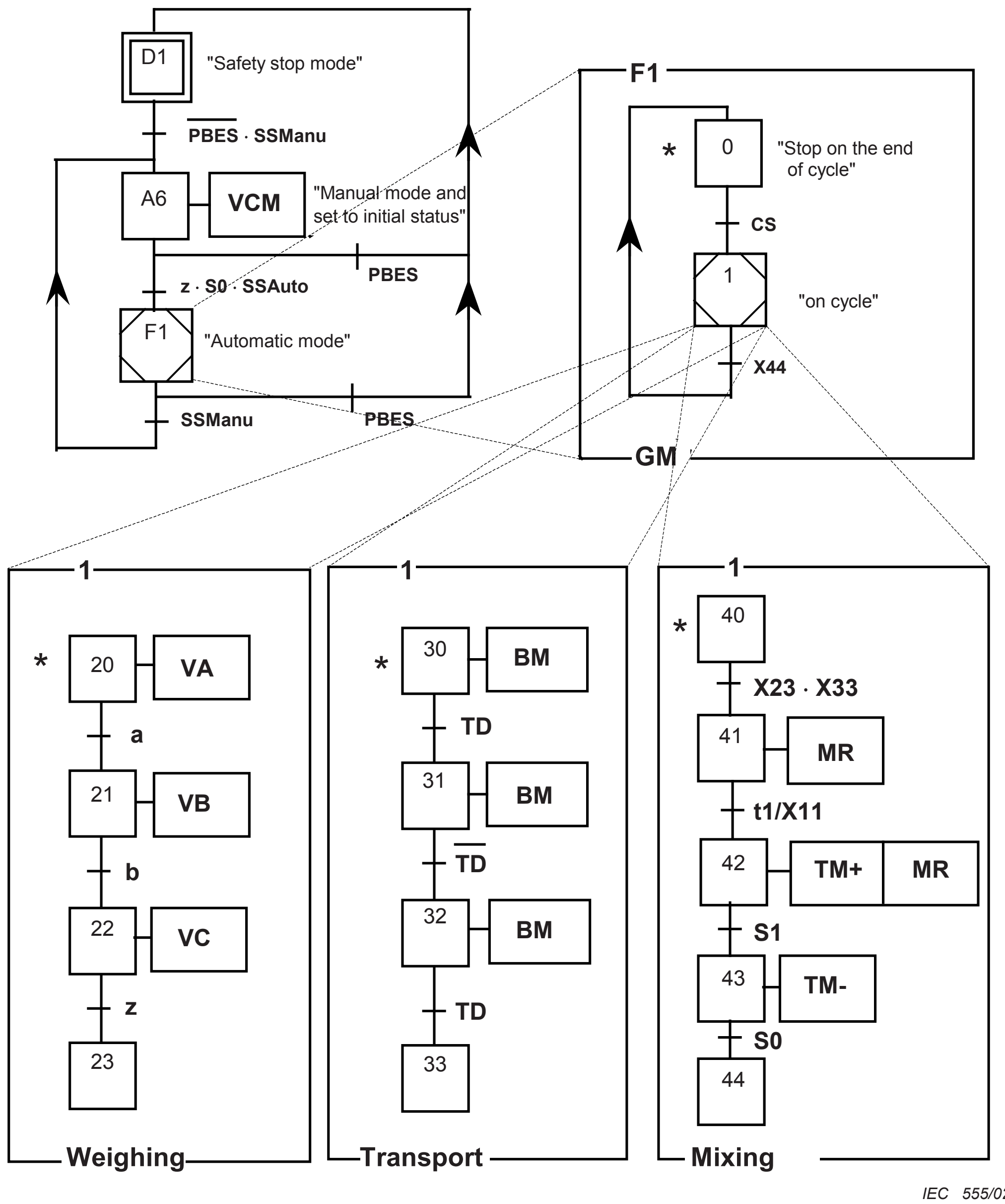
The forcing hierarchy involves two levels.

The forcing order, associated with the step D1 of the partial grafcet 1, forces the partial grafcet 10 to the empty situation (none of the steps of G10 is active).

The forcing order, associated with the step A6 of G1, forces G10 to the situation in which step 1 alone is active (but no transition is clearable).

The activation of the step F1 allows the normal evolution of G10 (because it does not depend of a forcing order).

Figure B.5 – Structuring with operating modes using forcing orders



The enclosure hierarchy involves three levels.

When step D1 is active, no other step is active.

When step A6 is active, no other step is active and the manual controls (EMC) are enabled.

When the enclosing step F1 is activated, step 0 of its enclosure GM is also activated.

When the enclosing step 1 is activated, step 20 of its "weighing" enclosure, step 30 of its "transport" enclosure and step 40 of its "mixing" enclosure are also activated.

Figure B.6 – Structuring with operating modes using enclosing step

Annex C
(informative)

Relations between GRAFCET of IEC 60848 and the SFC of IEC 61131-3

C.1 Introduction

IEC 60848 and IEC 61131-3 each have a specific domain of application:

- a behaviour specification language (GRAFCET – GRAphe Fonctionnel de Commande Etape Transition) independent of any specific technology of implementation, for IEC 60848, and
- a specific programming language (SFC – Sequential Function Chart), for IEC 61131-3.

GRAFCET of IEC 60848 is used by a grafcet chart to describe/specify the behaviour of system, as viewed from "outside" of the system, while the SFC language of IEC 61131-3 is used to describe (part of) the implemented software structure "inside" of the system.

If the two languages were both used to describe a control system, the two descriptions (two different document kinds) would in a given case look graphically similar. However, they would not have the same meaning, not even if they were graphically identical. This would just indicate that the structure of the software program, described in a software diagram, behaves in a way such that it can be described with a graphically similar grafcet chart. The properties of the underlying elements associated with the graphical element representations are nevertheless different in the two cases.

A specification using IEC 60848:2012 needs to be interpreted before implementation as a program using IEC 61131-3. There is presently no textual representation available for IEC 60848:2012 to support the interpretation and possible conversion into a program for an automation system.

C.2 In detail

a) GRAFCET designates a language of specification of the logical behaviour of systems. This specification is independent of the technology of realisation considered.

b) The SFC designates one of the programming languages defined in IEC 61131-3:1993. This language is inspired by IEC 60848:1988, but there is no identity between the two graphic representations and the semantics of the two languages.

c) At the present time, no method is known to translate a GRAFCET specification language into a SFC program: it is necessary to translate the theoretical semantics of the first in the implemented semantics of the other. That is why, whereas more synthetic and more ergonomic, GRAFCET is generally used for a global specification and the SFC for the detailed conception.

d) The revision of IEC 60848:1988, while preserving the existing syntax from the first edition and independence vis-à-vis the realisations, has been improved on the following points, allowing:

   - a better definition of elements and rules of evolution (internal events, distinction between assignation and allocation, transient evolution, etc.);
   - a finer interpretation of the behaviour specified, resorting to a principle or a realisation algorithm to understand the evolution in the difficult cases is therefore not required;
   - the emergence of help tools for the conception, capable of validating the specification notably by the calculation of property proofs;
   - the emergence of help tools for the realisation, capable of a guaranteed translation in diagrams or languages adapted to the technologies chosen;

- a better definition of structuring means (macro-steps, enclosing steps, etc.) associated to the GRAFCET language, capable of supporting the use of effective conception methods;
- a better formalisation of interfaces (predicate, allocation) brings in the Boolean variables manipulated by GRAFCET and the other variable types which are necessary to complete the description of the behaviour of the target system.

e) The search for a systematic identity between a GRAFCET element and its SFC corresponding element leads to the imposition of the programmed semantics of the second language into the first one. This approach would limit the role of the GRAFCET standard, IEC 60848:1988, solely to a general definition of symbols and rules intended to sustain definitions of the SFC elements of IEC 61131-3:1993.

f) IEC 61131-3:1993 specifies programming languages especially adapted to the PLCs (Programmable Logic Controllers). That is why it seems unlikely that this programmed description of the behaviour can apply in other technological contexts (electromechanical, electronic, pneumatic, etc.).

g) The evolution of distributed automatic devices, pre-wired and/or pre-programmed, creates the need to describe, in a formal and ergonomic way, the behaviour (guaranteed by the manufacturer) of these new devices in a language facilitating their integration in an automated application.

C.3 Possible future evolution

Facing other non-standardized candidates (for example the state chart), the GRAFCET notation benefits from an experience acquired by a big number of designers of automatic machines. This advantage should be used through an evolution of the standard to widen the domain of this language and to make it more formal and more structured.

A future integrated approach has been discussed by IEC subcommittee 3B and IEC Technical Committee 65 allowing a textual output of IEC 60848:1988 to be automatically converted into a structure and a program following IEC 61131-3:1993 and IEC 61499 respectively.

Bibliography

IEC 60617, *Graphical symbols for diagrams*. Available from http://std.iec.ch/iec60617

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™