INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Z.106

(10/96)

SERIES Z: PROGRAMMING LANGUAGES

Specification and Description Language (SDL)

Common interchange format for SDL

ITU-T Recommendation Z.106

(Previously CCITT Recommendation)

ITU-T Z-SERIES RECOMMENDATIONS

PROGRAMMING LANGUAGES

Specification and Description Language (SDL)	**Z.100–Z.109**
Criteria for the use and applicability of formal Description Techniques	Z.110–Z.199
ITU-T High Level Language (CHILL)	Z.200–Z.299
MAN-MACHINE LANGUAGE	Z.300–Z.499
General principles	Z.300–Z.309
Basic syntax and dialogue procedures	Z.310–Z.319
Extended MML for visual display terminals	Z.320–Z.329
Specification of the man-machine interface	Z.330–Z.399
Miscellaneous	Z.400–Z.499

For further details, please refer to ITU-T List of Recommendations.

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation Z.106 was prepared by ITU-T Study Group 10 (1993-1996) and was approved by the WTSC (Geneva, 9-18 October 1996).

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

SUMMARY

This Recommendation defines the Common Interchange Format (CIF) of CCITT's Specification and Description Language (Recommendation Z.100 – SDL). The CIF is intended for the interchange of graphical SDL specifications (SDL-GR) made on different tools that do not use the same storage format. Currently, the textual representation of SDL (SDL-PR) is used to interchange specifications with the disadvantage that all graphical information is lost making the same specifications often look very dissimilar in different environments. With the CIF, this disadvantage is reduced to a minimum, as it contains most of the graphical information. The CIF will improve the independence from specific tool vendors and will allow standard bodies to accept specifications in SDL-CIF irrespective of the tool they use for their internal work. This will also improve productivity by allowing specifications to be made on the accustomed tool. All SDL tool vendors are encouraged to provide facilities for importing and exporting SDL-CIF.

This Recommendation defines how SDL descriptions can be stored in order to be interchanged between tools coming from different vendors. It does not take into account the MSC notation. SDL-CIF is an extension to SDL-PR and is based on the SDL-PR syntax and can be read and written by tools as well as users. All the constructs available in SDL-PR are available in SDL-CIF with the exception of the macro call construct. As a result, most SDL-PR descriptions are legal SDL-CIF descriptions. SDL-CIF is an open storage format as it includes a mechanism of tool-specific directives. This mechanism allows a CIF-compliant tool to extend the format by adding specific information. SDL-CIF is also easily implementable and provides tool vendors with two levels of tool conformance and concepts of mandatory and optional directives.

This Recommendation first introduces SDL-CIF. Two conformance levels are defined, one at the SDL-PR level and the second including graphical information. Then the complete grammar is described with the related semantics. Mandatory and optional directives are described, as well as the format for tool-specific directives. Current tool-specific directives are described in Appendix I.

Two levels of CIF conformance are defined as level 1 and level 2. Level 1 is very close to SDL-PR, but it supports incomplete SDL specifications. Level 2 includes level 1 and is able to capture most of the graphical information of SDL-GR diagrams. A CIF specification must identify which of the two levels it complies with. Similarly, tools vendors that use the CIF should also identify the CIF level they comply with for their import and export functions.

BACKGROUND

Since a number of years, the Specification and Description Language (SDL) has been increasingly used, both in industry and for standards and Recommendations. Whereas in industry, SDL is often used in an environment with a single SDL tool, environments for standards and Recommendation creation often require the integration of SDL specifications from many tools used by different organizations. This is often also a requirement in international projects.

Until the time this Recommendation has been proposed, the only way to interchange specifications in SDL has been to interchange SDL-PR (the textual representation of SDL) Recommendations. This has led to the loss of graphical information. Though not necessary from the formal point of view, graphical information often has had a significant impact on readability and comprehensibility. With the Common Interchange Format, this Recommendation fulfils a long-expressed need for the interchange of SDL specifications without the loss of graphical information.

Recommendation Z.106

COMMON INTERCHANGE FORMAT FOR SDL

(Geneva, 1996)

1 Scope

This Recommendation defines the Common Interchange Format for specifications written in CCITT's Specification and Description Language (SDL) [1]. It is intended for tool vendors as an export and import format to allow the interchange of SDL specifications with tools offered by other tool vendors. The version described here does not cover new features being discussed for the 1996 SDL version and does not include Z.105 additions. Even though the format allows writing specifications in CIF directly, it is not intended for this purpose but rather should be generated by an existing SDL specification in the graphical representation (SDL-GR).

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

[1] ITU-T Recommendation Z.100 (1993), *CCITT Specification and Description Language (SDL)*.

3 Abbreviations

This Recommendation uses the following abbreviations.

CIF Common Interchange Format

SDL CCITT Specification and Description Language

4 Conventions, notation used

This notation is used in the grammar that follows:

[X]	X is optional.
{ X Y }	X and Y are together considered as one element.
X *	X is used zero or more times.
X +	X is used one or more times.
X | Y	either X or Y is used.
<yyy: Ax>	is a reference to another CIF rule. yyy is the name of the rule. Ax is the number of the rule.
<yyy: Bx>	is a reference to another CIF rule. yyy is the name of the rule. Bx is the number of the rule.
YYY	is a terminal symbol. YYY are the characters used for the terminal symbol. Case is not considered, i.e. yyy is the same as YYY.
<yyy>	is a PR construct taken from Recommendation Z.100. They are not a part of the SDL-CIF grammar but duplicated here for convenience.
<beginning of yyy>	is a part of a PR construct taken from Recommendation Z.100.
<end of yyy>	is a part of a PR construct taken from Recommendation Z.100.
<enhanced yyy>	is a PR construct taken from Recommendation Z.100 with embedded comments.

5 Level 1 CIF (CIF/PR)

5.1 General principles

CIF level 1 is a relaxation of SDL-PR syntax: it does not bring any additional information about graphical presentation, it only enhances the suitability of SDL-PR as an interchange format.

It overcomes the main drawback of SDL-PR as an interchange format, which is that SDL-PR only describes syntactically complete SDL descriptions, while there is a need to interchange descriptions which are partial or not yet achieved. However these descriptions must be syntactically correct to be interchanged.

SDL-CIF reuses large parts of the SDL-PR syntax. The production rules of SDL-PR which are reused in SDL-CIF are later just referenced by their name, they are not described again in this Recommendation.

The Z.100 rules to transform SDL-PR into the abstract grammar also apply to parts of SDL-CIF which are shared with SDL-PR, as far as it is possible according to the incompleteness of the SDL-CIF descriptions.

5.2 Transferable units of SDL specifications

5.3 CIF/PR syntax

5.3.1 CIF file

The starting production rule of Recommendation Z.100, <sdl specification> (see 2.4.1.1/Z.100), is replaced by the following production:

<cif level 1 file> ::=

{ <package definition>

| <textual system definition>

| <definition>

} *

The productions <package definition>, <textual system definition> and <definition> are identical to the matching Z.100 SDL-PR productions.

Reminder:

<definition (2.4.1.3/Z.100)> ::=

<system type definition>

| <block definition>

| <block type definition>

| <process definition>

| <process type definition>

| <service definition>

| <service type definition>

| <procedure definition>

| <block substructure definition>

| <channel substructure definition>

| <macro definition>

| <operator definition>

5.3.2 Macro call

SDL-PR macro calls may "appear at any place where a <lexical unit> is allowed" (4.2.3/Z.100).

SDL-CIF macro calls can only appear at the place of a task symbol.

The production rule <action statement> (see 2.6.8.1/Z.100) is replaced by:

<action statement> ::=

[<label>] { <action> <end> | <macro call> }

5.4 Examples

Example 1

```
process p;
start; stop;
endprocess;
```

6 Level 2 CIF (CIF/GR)

6.1 General principles

CIF level 2 is an extension of CIF level 1 with so-called "CIF directives" that describe the main characteristics of the graphical representation of objects.

CIF directives are placed before the object they are associated with. One design principle used when defining this Recommendation has been that: All SDL-PR constructs that contain information that the CIF converter has to extract should have an associated CIF comment placed before the SDL-PR construct. This makes it possible for a CIF reading tool to scan for the next CIF comment, extract necessary information from the following SDL-PR and then start looking for the next CIF comment.

Several CIF directives may be associated with the same object.

The first CIF directive for an object usually describes the layout of the main part of the object, while the following CIF directives for the same object describe the layout of subparts of the object.

CIF level 2 does not describe all the details of the graphical representation, as this would restrict too much the number of tools able to support SDL-CIF: some SDL editors favour manual (user) layout of symbols while others favour automatic layout. Handling both manual and automatic layout is a complex problem which is difficult to solve when developing an SDL tool.

In order to face this issue, CIF directives are classified in three categories: mandatory, optional and tool-specific directives.

Mandatory directives describe graphical characteristics which cannot be automatically computed, or whose automatic computation would almost certainly be too far from the user expectations, e.g. the layout of symbols and lines in interconnection diagrams.

These graphical characteristics are generally user-controlled in the main SDL editing tools.

Optional directives describe graphical characteristics which can be automatically computed, for instance text layout inside symbols. For any optional information which is not given, tools should automatically compute a layout.

Tool-specific directives describe characteristics (graphical or not) which are not covered by mandatory or optional directives. They allow tool manufacturers to add new CIF directives in the storage format which will be analysed by their own tools only.

6.2 General principles, graphical information

6.2.1 The coordinate system

The unit used is 1/10 mm. Origo is the upper left corner of a page. The positive x-axis is to the right of origo. The positive y-axis is below origo.

Pages

Diagrams may be split into pages, as described in 2.2.5/ Z.100.

However SDL-CIF files are not structured according to pages, but according to the SDL-PR syntax. Pages are described by CIF comments inserted between some syntax units. A page may consist of information from several non-adjacent syntax units.

Pages are independent drawing areas. Every coordinate must be interpreted according to the current page name.

Classification of information

CIF graphical information can be classified into 4 classes:

- information about symbols which look like graphical lines, usually just called "lines", i.e. signal routes, channels, flow lines in transitions, and association lines;
- information about symbols which do not look like lines, usually just called "symbols", e.g. process symbols, output symbols;
- information about text;
- other information, e.g. page splitting information.

Symbol representation

All information about symbol positions and sizes is mandatory information.

The position of a symbol is usually given by the coordinates of the upper left corner of its bounding box. There are a few exceptions, where the upper right corner is used instead (for reversed text extension symbols and reversed comment symbols).

The size of a symbol is given by the width and height of its bounding box.

Symbol-specific information is sometimes added. One example is that for symbols which exist in both a left and right version, a "Left" or "Right" keyword may be present.

Text representation

Text positions and sizes are optional information. They refer to the text bounding box and not the text itself.

6.2.2 About optional text positions

The specification of different kinds of text positions is optional in CIF. This means that a tool does not have to specify a text position when writing a CIF file. It does also mean that a tool reading a CIF file with a specified text position does not have to use that text position. Here are some guidelines:

When reading a CIF file, a tool should try to use the text positions found in the CIF file. If this is not possible, use autolayout instead.

If a text position is not given in the CIF file, autolayout should be used.

Some text positions are more important than others to retain the original SDL-GR layout of a diagram in CIF. A tool maker should concentrate on implementing support for these text positions first. Below, text positions are listed in groups. Text positions that are most important to preserve are in the first group.

- Group 1: channel name, signal list, signal route name, gate name, select.
- Group 2: connect, answer flow line, gate reference, return.
- Group 3: diagram kernel heading, page name, system symbol, block symbol, process symbol, service symbol, procedure symbol, operator symbol, state, save, task, set, reset, create, procedure call, procedure start, decision, continuous signal, enabling condition, transition option, join, label, macro call, macro outlet, input, priority input, output, text, package reference.

Line representation

The layout of lines is given by a list of coordinates: one for the start point of the line, one for every break point on the line, and one for the end point of the line.

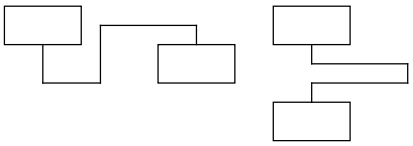
The start and end points of a line are usually connected to other symbols.

If a start or end point is connected to a symbol that does not have the shape of a rectangle, the point is on the symbol's bounding box instead of on the real symbol outline, to simplify geometry computations.

The layout of channel and signalroute lines is a mandatory information, as it is impossible to guess what the user wants to see.

6.2.3 About optional flow lines

Flow lines not following directly after a decision symbol are optional in the same sense as text positions. Some guidelines:



It is most important to give information about complicated flow lines like the two flowlines above.

It is less important to give information about simple flow lines like the two flowlines above.

T1008850-96/d01

Graphical information not covered by CIF

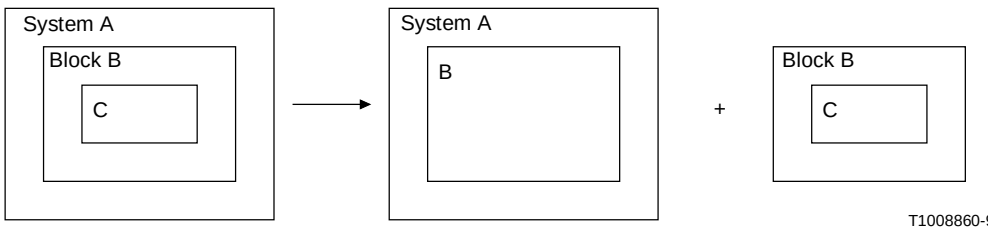
CIF directives are mainly related to graphical positions and sizes, because it is a universal information that can be interchanged without implementation problems, and because it is information that would make diagrams very difficult to redraw if it was missing.

Some other kinds of information have been agreed as less important, and are not covered by SDL-CIF. These are information about text font, font size, colour and line thickness.

They can be given by tool-specific directives.

6.2.4 About nested diagrams

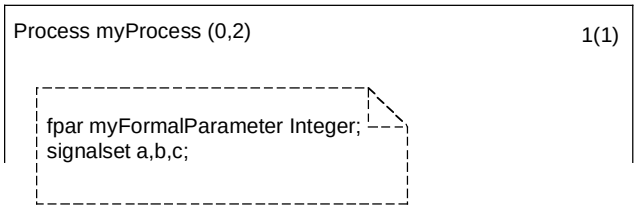
Nested SDL (diagrams within diagrams) is supported by CIF. Tools that do not support nested SDL can convert a diagram within a diagram to a reference symbol and a separate diagram:



T1008860-96/d02

6.2.5 About kernel and additional heading

CIF does not support additional headings. This means that tools supporting additional headings have to make the partitioning of the heading text into a kernel heading and an additional heading without support from CIF. Note that the SDL-GR in the example below is not correct, because according to Recommendation Z.100 there should not be a symbol around the additional heading text.



T1008870-96/d03

A heading text split into kernel heading and additional heading.

6.3 CIF/GR lexical rules

6.3.1 CIF directives

CIF directives are special forms of the Z.100 <note> comments, all of which have in common the following description:

<cif directive> ::=

/* { CIF | cif } <text> */

A source line in the analysed file, which contains a CIF directive, must not contain any other token.

A CIF/GR <note> must not be a <cif directive>.

6.3.2 Newline and space characters

Newline and space characters are usually considered as non-significant characters when encountered during the analysis of the CIF file, and are then ignored.

However when two adjacent SDL tokens are displayed in a diagram as adjacent parts of a text object, the newline and space characters between the two tokens should not be ignored: they should be used as a part of the text object.

This allows tools to keep the user preference for the text layout.

When two SDL tokens of a text object are separated by some space characters and a newline followed by several space characters, the space characters before the first significant character of the second line must be ignored as they are indentation spaces.

The first significant space character of a line is the character which is at the same column as the first '/' character of the previous CIF directive.

For example, in the following SDL-CIF fragment:

```
BLOCK b;
   /* CIF Signalroute (500,400),(300,400) */
   SIGNALROUTE r FROM ENV To P WITH s1   ,   s2
      s3;
```

The text which must be displayed for the list of signals is:

```
's1   ,   s2' // NL // '  s3'
```

(provided there are no space characters after "s2").

6.3.3 About text layout

The placement of newline characters in SDL-GR should be preserved in SDL-PR, i.e. a signal list in GR with two signals on two lines should have the PR:

```
FROM ENV TO P WITH KeyStroke,
Card;
```

instead of:

```
FROM ENV TO P WITH KeyStroke, Card;
```

6.4 CIF/GR syntax

6.4.1 CIF A Rules

A normal rule (see for instance <start symbol: A43>) is in general described like this:

- A section describing the grammar for the CIF comment and for related SDL-PR information. This section also shows how CIF comments should be placed in the SDL-PR code.
- A section with cross reference information listing all rules using this rule.
- A section giving details of related SDL-PR constructs taken directly from Z.100: <xxx> is ...
- A section with explaining text and an example.

Note that this grammar only gives instructions on how to insert CIF comments into SDL-PR. A CIF file is only correct if it is based on a correct SDL-PR fragment.

A1 CIF description:

{ <diagram description: A2> }*

Additional information:

This is the start rule.

A2 diagram description:

<diagram start: A3> { <CIF descriptor: A18> }* [<diagram end: A17>]

This rule is used by <CIF description: A1>, <CIF descriptor: A18>.

Additional information:

<diagram end: A17> should be given if the diagram is not a typebased system diagram (i.e. if not <textual typebased system definition> is used in SDL-PR), see <system diagram start: A5>.

A3 diagram start:

<package diagram start: A4> | <system diagram start: A5> | <system type diagram start: A6> | <block diagram start: A7> | <block type diagram start: A8> | <substructure diagram start: A9> | <process diagram start: A10> | <process type diagram start: A11> | <service diagram start: A12> | <service type diagram start: A13> | <procedure diagram start: A14> | <operator diagram start: A15> | <macro diagram start: A16>

This rule is used by <diagram description: A2>.

A4 package diagram start:

/* CIF PackageDiagram */
{ <page declaration: B3> }+
<beginning of package definition>

This rule is used by <diagram start: A3>.

<beginning of package definition> is
{<package reference clause>}* **PACKAGE** <name> [<interface>] <end>

<package reference clause> is
USE <package name> [/ <definition selection list>] <end>

<definition selection list> is
<definition selection> { **,** <definition selection> }*

<definition selection> is
[<entity kind>] <name>

<entity kind> is (SDL92)
{ **SYSTEM TYPE** } | { **BLOCK TYPE** } | { **PROCESS TYPE** } | { **SERVICE TYPE** } | **SIGNAL** | **PROCEDURE** | **NEWTYPE** | **SIGNALLIST** | **GENERATOR** | **SYNONYM** | **REMOTE**

<entity kind> is (SDL96)
{ **SYSTEM TYPE** } | { **BLOCK TYPE** } | { **PROCESS TYPE** } | { **SERVICE TYPE** } | **SIGNAL** | [**REMOTE**] **PROCEDURE** | **NEWTYPE** | **SIGNALLIST** | **GENERATOR** | **SYNONYM** | **REMOTE**

<interface> is
INTERFACE <definition selection list>

Additional information:

There should be one <page declaration: B3> for each page in the diagram.

Example:


```
/* CIF PackageDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,250),(1700,1950) */
/* CIF PackageReference (125,25) */
use myOtherPackage;
Package myPackage;
```

A5 system diagram start:

/* CIF SystemDiagram */
{ <page declaration: B3> }+
<enhanced textual system definition>

This rule is used by <diagram start: A3>.

<enhanced textual system definition> is
{ <package reference clause> }* { **SYSTEM** <name> <end> |
<textual typebased system definition> /* CIF End SystemDiagram */ }

Additional information:

There should be one <page declaration: B3> for each page in the diagram.

Example 1:

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
SYSTEM mySystem;
```

Example 2:

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
SYSTEM mySystem : mySystemType;
/* CIF End SystemDiagram */
```

A6 system type diagram start:

/* CIF SystemTypeDiagram */
<diagram parts: B1>
<beginning of system type definition>

This rule is used by <diagram start: A3>.

<beginning of system type definition> is
SYSTEM TYPE { <name> | <identifier> } [<formal context parameters>] [<specialization>] <end>

<identifier> is
[<qualifier>] <name>

<qualifier> is
{ <path item> { / <path item> }* } |
{ << <path item> { / <path item> }* >> }

<path item> is (SDL92)
<scope unit kind> { <name> | <quoted operator> }

<path item> is (SDL96)
<scope unit kind> { <name> | <quoted operator> | <operator name> <exclamation> }

<scope unit kind> is
PACKAGE | { **SYSTEM TYPE** } | **SYSTEM** | **BLOCK** | { **BLOCK TYPE** } | **SUBSTRUCTURE** | **PROCESS** | { **PROCESS TYPE** } | **SERVICE** | { **SERVICE TYPE** } | **PROCEDURE** | **SIGNAL** | **OPERATOR** | **TYPE**

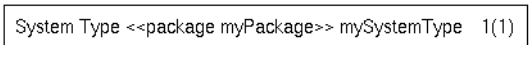
<quoted operator> is
{ <quote> <infix operator> <quote> } | { <quote> <monadic operator> <quote> }

<quote> is "

<formal context parameters> is ...

<specialization> is
INHERITS <type expression> [**ADDING**]

Example:




```
/* CIF SystemTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
System Type <<package myPackage>> mySystemType;
```

A7 block diagram start:

/* CIF BlockDiagram */
<diagram parts: B1>
<beginning of block definition>

This rule is used by <diagram start: A3>.

<beginning of block definition> is
BLOCK { <name> | <identifier> } <end>

Example:


```
/* CIF BlockDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Block myBlock;
```

A8 block type diagram start:

/* CIF BlockTypeDiagram */
<diagram parts & gate references: B2>
<beginning of block type definition>

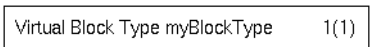
This rule is used by <diagram start: A3>.

<beginning of block type definition> is
[<virtuality>] **BLOCK TYPE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>]
[<specialization>] <end>

<virtuality> is
VIRTUAL | **REDEFINED** | **FINALIZED**

<virtuality constraint> is
ATLEAST <identifier>

Example:




```
/* CIF BlockTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Virtual Block Type myBlockType;
```

A9 substructure diagram start:

/* CIF SubstructureDiagram [**Invisible**] ***/**
[<diagram parts & gate references: B2>]
{ <beginning of block substructure definition> |
<beginning of channel substructure definition> }

This rule is used by <diagram start: A3>.

<beginning of block substructure definition> is
SUBSTRUCTURE { <name> | <identifier> } <end>

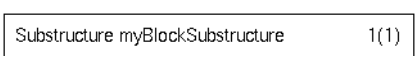
<beginning of channel substructure definition> is
SUBSTRUCTURE { <name> | <identifier> } <end>

Additional information:

The substructure diagram is **Invisible** if the substructure diagram is not visible in SDL-GR. This is the case when the GR shorthand block in block is used.

If **Invisible** is given, <diagram parts & gate references: B2> should be omitted. If **Invisible** is omitted, <diagram parts & gate references: B2> should be present. Read more about invisible substructure diagram in the examples document.

Example:




```
/* CIF SubstructureDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Substructure myBlockSubstructure;
```

A10 process diagram start:

/* CIF ProcessDiagram */
<diagram parts: B1>
<beginning of process definition>

This rule is used by <diagram start: A3>.

<beginning of process definition> is
PROCESS { <name> | <identifier> } [<number of process instances>] <end> [<formal parameters> <end>]
[<valid input signal set>]

<number of process instances> is
([<initial number>] [**,** [<maximum number>]] **)**

<formal parameters> is
FPAR <parameters of sort> { **,** <parameters of sort> }*

<parameters of sort> is
<variable name> { **,** <variable name> }* <sort>

<sort> is
<sort identifier> | <syntype>

<syntype> is
<syntype identifier>

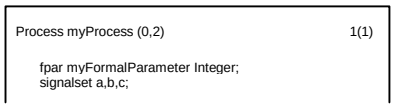
<valid input signal set> is
SIGNALSET [<signal list>] <end>

<signal list> is
<signal list item> { **,** <signal list item> }*

<signal list item> is (SDL92)
<signal identifier> | **(** <signal list identifier> **)** | <timer identifier>

<signal list item> is (SDL96)
<signal identifier> | **(** <signal list identifier> **)** | <timer identifier> | [**PROCEDURE**] <remote procedure identifier> |
[**REMOTE**] <remote variable identifier>

Example:




```
/* CIF ProcessDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Process myProcess(0,2);
fpar myFormalParameter Integer;
signalset a,b,c;
```

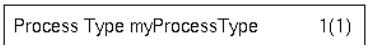
A11 process type diagram start:

/* CIF ProcessTypeDiagram */
<diagram parts & gate references: B2>
<beginning of process type definition>

This rule is used by <diagram start: A3>.

<beginning of process type definition> is
[<virtuality>] **PROCESS TYPE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>]
[<specialization>] <end> [<formal parameters> <end>] [<valid input signal set>]

Example:




```
/* CIF ProcessTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Process Type myProcessType;
```

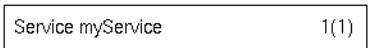
A12 service diagram start:

/* CIF ServiceDiagram */
<diagram parts: B1>
<beginning of service definition>

This rule is used by <diagram start: A3>.

<beginning of service definition> is
SERVICE { <name> | <identifier> } <end> [<valid input signal set>]

Example:




```
/* CIF ServiceDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Service myService;
```

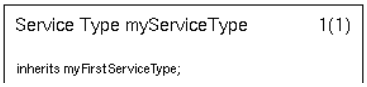
A13 service type diagram start:

/* CIF ServiceTypeDiagram */
<diagram parts: B1>
<beginning of service type definition>

This rule is used by <diagram start: A3>.

<beginning of service type definition> is
[<virtuality>] **SERVICE TYPE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>]
[<specialization>] <end> [<valid input signal set>]

Example:




```
/* CIF ServiceTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Service Type myServiceType inherits myFirstServiceType;
```

A14 procedure diagram start:

/* CIF ProcedureDiagram */
<diagram parts: B1>
<beginning of procedure definition>

This rule is used by <diagram start: A3>.

<beginning of procedure definition> is (SDL92)
<procedure preamble> **PROCEDURE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>] [<specialization>] <end> [<procedure formal parameters> <end>] [<procedure result> <end>]

<beginning of procedure definition> is (SDL96)
<external procedure definition> |
<procedure preamble> **PROCEDURE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>] [<specialization>] <end> [<procedure formal parameters> <end>] [<procedure result> <end>]

<external procedure definition> is (SDL96)
PROCEDURE <procedure name> [<procedure signature>] **EXTERNAL** <end>

<procedure signature> is
[[<end>] **FPAR** <procedure formal parameter constraint> { , <procedure formal parameter constraint> } * [<end>
RETURNS <sort>]] |
[<end>] **RETURNS** <sort>

<procedure formal parameter constraint> is
<parameter kind> <sort>

<procedure preamble> is
[<virtuality>] [**EXPORTED** [**AS** <identifier>]]

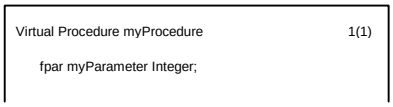
<procedure formal parameters> is
FPAR <formal variable parameters> { **,** <formal variable parameters> }*

<formal variable parameters> is
<parameter kind> <parameters of sort>

<parameter kind> is
[**IN/OUT** | **IN**]

<procedure result> is
RETURNS [<variable name>] <sort>

Example:




```
/* CIF ProcedureDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Virtual Procedure myProcedure;
fpar myParameter Integer;
```

A15 operator diagram start:

/* CIF OperatorDiagram */
<diagram parts: B1>
<beginning of operator definition>

This rule is used by <diagram start: A3>.

<beginning of operator definition> is (SDL92)
OPERATOR { <name> | <identifier> } <end> <formal parameters> <end> <operator result> <end>

<beginning of operator definition> is (SDL96)
<external operator definition> |
OPERATOR { <name> | <identifier> } <end> [<formal parameters> <end>] <operator result> <end>

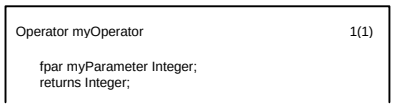
<external operator definition> is (SDL96)
OPERATOR <operator name> [<operator signature>] **EXTERNAL** <end>

<operator signature> is
<operator name> **:** <argument list> **->** <result> |
ORDERING |
NOEQUALITY

<operator result> is (SDL92)
RETURNS [<variable name>] <extended sort>

<operator result> is (SDL96)
RETURNS [<variable name>] <sort>

Example:




```
/* CIF OperatorDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Operator myOperator;
fpar myParameter Integer;
returns Integer;
```

A16 macro diagram start:

/* CIF MacroDiagram */
{ <page declaration: B3> }+
<beginning of macro definition>

This rule is used by <diagram start: A3>.

<beginning of macro definition> is
MACRODEFINITION <name> [<macro formal parameters>] <end>

<macro formal parameters> is
FPAR <macro formal parameter> { **,** <macro formal parameter> }*

<macro formal parameter> is
<name>

Additional information:

General macro diagrams are not supported by CIF. Macros with one inlet and one outlet symbol that are able to replace a task symbol in a process are supported by this rule, <macro call symbol: A61>, <macro inlet symbol: A62> and <macro outlet symbol: A63>.

There should be one <page declaration: B3> for each page in the diagram.

Example:

```
/* CIF MacroDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
MACRODEFINITION myMacro;
```

A17 diagram end:

{ /* **CIF End PackageDiagram** */ <end of package definition> |
/* **CIF End SystemDiagram** */ <part of end of textual system definition> |
/* **CIF End SystemTypeDiagram** */ <end of system type definition> |
/* **CIF End BlockDiagram** */ <end of block definition> |
/* **CIF End BlockTypeDiagram** */ <end of block type definition> |
/* **CIF End SubstructureDiagram** */ { <end of channel substructure definition> | <end of block substructure definition> } |
/* **CIF End ProcessDiagram** */ <end of process definition> |
/* **CIF End ProcessTypeDiagram** */ <end of process type definition> |
/* **CIF End ServiceDiagram** */ <end of service definition> |
/* **CIF End ServiceTypeDiagram** */ <end of service type definition> |
/* **CIF End ProcedureDiagram** */ <end of procedure definition> |
/* **CIF End OperatorDiagram** */ <end of operator definition> |
/* **CIF End MacroDiagram** */ <end of macro definition> }

This rule is used by <diagram description: A2>.

<end of package definition> is
ENDPACKAGE [<package name>] <end>

<part of end of textual system definition> is
ENDSYSTEM [<system name>] <end>

<end of system type definition> is
ENDSYSTEM TYPE [<system type name> | <system type identifier>] <end>

<end of block definition> is
ENDBLOCK [<block name> | <block identifier>] <end>

<end of block type definition> is
ENDBLOCK TYPE [<block type name> | <block type identifier>] <end>

<end of channel substructure definition> is
ENDSUBSTRUCTURE [{ <channel substructure name> | <channel substructure identifier> }] <end>

<end of block substructure definition> is
ENDSUBSTRUCTURE [{ <block substructure name> | <block substructure identifier> }] <end>

<end of process definition> is
ENDPROCESS [<process name> | <process identifier>] <end>

<end of process type definition> is
ENDPROCESS TYPE [<process type name> | <process type identifier>] <end>

<end of service definition> is
ENDSERVICE [{ <service name> | <service identifier> }] <end>

<end of service type definition> is
ENDSERVICE TYPE [{ <service type name> | <service type identifier> }] <end>

<end of procedure definition> is
ENDPROCEDURE [<procedure name> | <procedure identifier>] <end>

<end of operator definition> is
ENDOPERATOR [{ <operator identifier> | <operator name> }] <end>

<end of macro definition> is
ENDMACRO [<macro name>] <end>

Additional information:

This CIF comment is also used after a <textual typebased system definition>, see <system diagram start: A5>.

Example:

```
/* CIF End SystemTypeDiagram */
ENDSYSTEM TYPE mySystemType;
```

A18 CIF descriptor:

<diagram description: A2> | <default size: A19> | <page switch: A20> | <channel: A21> | <signal route: A22> | <gate: A23> | <connect: A24> | <create line: A27> | <flow line: A28> | <answer flow line: A29> | <block symbol: A30> | <dashed block symbol: A31> | <process symbol: A32> | <dashed process symbol: A33> | <service symbol: A34> | <dashed service symbol: A35> | <system type symbol: A36> | <block type symbol: A37> | <process type symbol: A38> | <service type symbol: A39> | <block substructure symbol: A40> | <procedure symbol: A41> | <operator symbol: A42> | <start symbol: A43> | <stop symbol: A44> | <state symbol: A45> | <nextstate symbol: A46> | <save symbol: A47> | <task symbol: A48> | <set symbol: A49> | <reset symbol: A50> | <export symbol: A51> | <create symbol: A52> | <procedure call symbol: A53> | <procedure start symbol: A54> | <return symbol: A55> | <decision symbol: A56> | <continuous signal symbol: A57> | <enabling condition symbol: A58> | <transition option symbol: A59> | <join symbol: A60> | <macro call symbol: A61> | <macro inlet symbol: A62> | <macro outlet symbol: A63> | <label symbol: A64> | <input symbol: A65> | <priority input symbol: A66> | <output symbol: A67> | <text symbol: A68> | <select symbol: A69> | <descriptor end: A70>

This rule is used by <diagram description: A2>.

Additional information:

<diagram description: A2> is used to describe nested diagrams.

A19 default size:

/* CIF DefaultSize <size point: B26> */

This rule is used by <CIF descriptor: A18>.

Additional information:

This comment may be placed before any symbol or line PR construct within a diagram.

The size given here will be used for all subsequent symbols without a defined size until a new default size is given. The default size is remembered even after a new <diagram start: A3>. It is illegal to omit a symbol size specification before a default size is given.

Example where the task symbol will get the size (200,100):

```
/* CIF DefaultSize (200,100) */
/* CIF Task (800,550) */
task GameP:=null;
```

A20 page switch:

/* CIF CurrentPage <page name> ***/**

This rule is used by <CIF descriptor: A18>.

<page number> is
<literal name>

Additional information:

This comment may be placed before any symbol or line CIF & PR construct within a diagram. The mentioned page becomes the current page. The page name must refer to a page declared in the previous diagram start CIF comment.

Everything after this CIF comment in this diagram will be placed on the current page, until another current page is defined. The current page for a diagram is initially set by a <page declaration: B3>.

Example:

```
/* CIF CurrentPage 1 */
/* CIF Task (800,550) */
task GameP:=null;
```

A21 channel:

/* CIF Channel <pointlist: B22> [**InvisibleName**] ***/**
[<channel name text position: B25>]
[<first signallist text position: B6>]
[<second signallist text position: B7>]
[<first arrow position: B8>]
[<second arrow position: B9>]
<enhanced channel definition>

This rule is used by <CIF descriptor: A18>.

<enhanced channel definition> is (SDL92)
CHANNEL <name> [**NODELAY**]
<channel path> [<channel path>]
[<channel substructure symbol: B4> <textual channel substructure reference>]
ENDCHANNEL [<name>] <end>

<enhanced channel definition> is (SDL96)
CHANNEL [<name>] [**NODELAY**]
<channel path> [<channel path>]
[<channel substructure symbol: B4> <textual channel substructure reference>]
ENDCHANNEL [<name>] <end>

<channel path> is (SDL92)
FROM <channel endpoint> **TO** <channel endpoint> **WITH** <signal list> <end>

<channel path> is (SDL96)
FROM <channel endpoint> **TO** <channel endpoint> [**WITH** <signal list>] <end>

<channel endpoint> is
{ <block identifier> | **ENV** } [**VIA** <gate name>]

<textual channel substructure reference> is
SUBSTRUCTURE <name> **REFERENCED** <end>

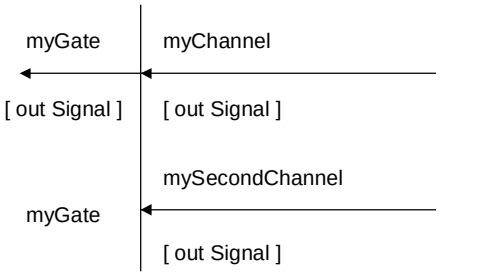
Additional information:

(SDL96) If **InvisibleName** is given, the channel name should not appear in GR. The name is only given in PR to be able to refer to it in a **CONNECT** statement. The **InvisibleName** keyword should not appear in CIF generated from SDL92.

The first point in the pointlist is on the surrounding rectangle of the symbol corresponding to the **FROM** <channel endpoint> of the first <channel path>. The last point in the pointlist is on the surrounding rectangle of the symbol corresponding to the **TO** <channel endpoint> of the first <channel path>.

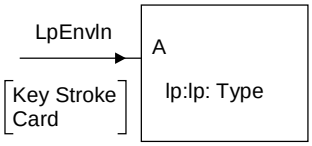
If a <first arrow position: B8> is not given, autolayout will be used for that arrow position when reading the CIF file.

The text position of the gate in the VIA construct is specified in either <gate: A23> or <gate reference: B19>. The picture below shows a gate and a reference to the gate.



T1008890-96/d05

Example 1:



T1008900-96/d06

```
/* CIF Block (500,350) */
/* CIF GateReference (500,400) */
/* CIF TextPosition (510,390) */
BLOCK lp:lpType;
/* CIF Channel (300,400),(500,400) */
/* CIF TextPosition (390,350) */
/* CIF TextPosition (390,410) SignalList1 */
CHANNEL LpEnvIn
FROM env TO Lp VIA A
WITH KeyStroke,
Card;
ENDCHANNEL LpEnvIn;
```

Example 2:

The example below describes a channel substructure 'myChannel' connected to a channel 'c'.

```
/* CIF Channel (700,400),(1100,400) */
/* CIF TextPosition (900,390) */
/* CIF TextPosition (700,350) SignalList1 */
/* CIF TextPosition (1100,350) SignalList2 */
CHANNEL c
FROM a TO b WITH s1;
FROM b TO a WITH s2;
/* CIF ChannelSubstructure (800,550) */
/* CIF Line (900,550),(900,400) Dashed */
SUBSTRUCTURE mySubstructure REFERENCED;
ENDCHANNEL c;
```

A22 signal route:

/* CIF SignalRoute <pointlist: B22> [**InvisibleName**] ***/**
[<signal route name text position: B25>]
[<first signallist text position: B6>]
[<second signallist text position: B7>]
<signal route definition>

This rule is used by <CIF descriptor: A18>.

<signal route definition> is (SDL92)
SIGNALROUTE <name>
<signal route path>
[<signal route path>]

<signal route definition> is (SDL96)
SIGNALROUTE [<name>]
<signal route path>
[<signal route path>]

<signal route path> is (SDL92)
FROM <signal route endpoint> **TO** <signal route endpoint> **WITH** <signal list> <end>

<signal route path> is (SDL96)
FROM <signal route endpoint> **TO** <signal route endpoint> [**WITH** <signal list>] <end>

<signal route endpoint> is
{ <process identifier> | <service identifier> | **ENV** } [**VIA** <gate>]

Additional information:

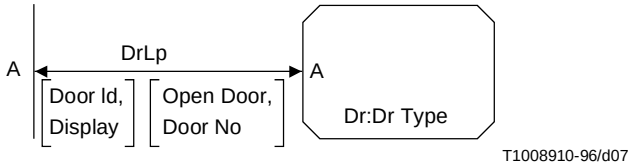
(SDL96) If **InvisibleName** is given, the signal route name should not appear in GR. The name is only given in PR to be able to refer to it from a **CONNECT** statement. The **InvisibleName** keyword should not appear in CIF generated from SDL92.

The first point in the pointlist is on the surrounding rectangle of the symbol corresponding to the **FROM** <signal route endpoint> of the first <signal route path>. The last point in the pointlist is on the surrounding rectangle of the symbol corresponding to the **TO** <signal route endpoint> of the first <signal route path>.

Arrows are placed with autolayout for a signal route when reading a CIF file.

The text position of the gate in the **VIA** construct is specified in either <gate: A23> or <gate reference: B19>.

Example:



T1008910-96/d07

```
/* CIF SignalRoute (500,400),(300,400) */
/* CIF TextPosition (390,410) */
/* CIF TextPosition (490,410) SignalList1 */
/* CIF TextPosition (290,410) SignalList2 */
SIGNALROUTE DrLp
FROM Dr VIA A TO env VIA A
WITH DoorId,
Display;
FROM env VIA A TO Dr VIA A
WITH OpenDoor,
DoorNo;
```

A23 gate:

/* CIF Gate <pointlist: B22> [**Dashed**] ***/**
[<gate name text position: B25>]
[<first signallist text position: B6>]
[<second signallist text position: B7>]
[<gate constraint symbol: B5>]
<gate definition>

This rule is used by <CIF descriptor: A18>.

<gate definition> is
GATE <gate name> [**ADDING**] <gate constraint> <end> [<gate constraint> <end>]

<gate constraint> is
{ **OUT** [**TO** <textual endpoint constraint>] | **IN** [**FROM** <textual endpoint constraint>] } [**WITH** <signal list>]

<textual endpoint constraint> is
[**ATLEAST**] <identifier>

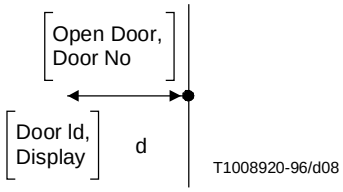
Additional information:

There should be two points in the <pointlist: B22>. The first point in the pointlist is on the surrounding rectangle of the frame symbol of the diagram. The second point in the pointlist should be the other point that defines the gate. If a <textual endpoint constraint> exists, the second point will be on the surrounding rectangle of the symbol corresponding to the <textual endpoint constraint>.

Dashed should be used if the keyword **ADDING** is used in SDL-PR.

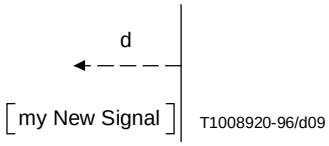
The <gate constraint symbol: B5> should be given if a <textual endpoint constraint> is given.

Example 1:



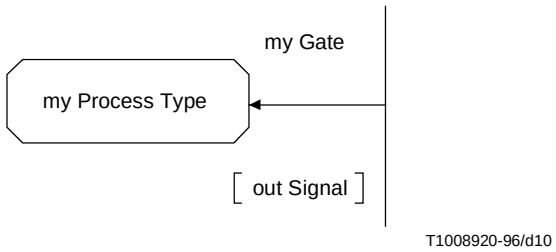

```
/* CIF Gate (500,400),(300,400) */
/* CIF TextPosition (390,410) */
/* CIF TextPosition (490,410) SignalList1 */
/* CIF TextPosition (290,410) SignalList2 */
GATE d OUT
WITH DoorId,
Display;
IN
WITH OpenDoor,
DoorNo;
```

Example 2:




```
/* CIF Gate (500,400),(300,400) Dashed */
GATE d ADDING OUT
WITH myNewSignal;
```

Example 3:




```
/* CIF Gate (500,400),(300,400) */
/* CIF Process (100,350),(200,100) */
GATE myGate OUT TO myProcessType
WITH outSignal;
```

A24 connect:

/*** CIF Connect */**
[<text position: B25>]
{ <channel to route connection> | <signalroute to route connection> | <channel connection> }

This rule is used by <CIF descriptor: A18>.

<channel to route connection> is
CONNECT <channel identifiers> **AND** <signal route identifiers> <end>

<channel identifiers> is
<channel identifier> { **,** <channel identifier> }*

<signal route identifiers> is
<signal route identifier> { **,** <signal route identifier> }*

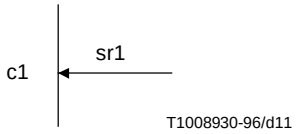
<signal route to route connection> is
CONNECT <external signal route identifiers> **AND** <signal route identifiers> <end>

<channel connection> is
CONNECT <channel identifiers> **AND** <subchannel identifiers> <end>

Additional information:

<text position: B25> is the text position for the <channel identifiers> or the <external signal route identifiers>, i.e. the text outside the frame symbol.

Example:




```
/* CIF Connect */
/* CIF TextPosition (800,50) */
CONNECT c1 AND sr1;
```

A25 text extension:

<text outside extension>
/* CIF TextExtension <position and size: B24>
[{ **Left** | **Right** }] ***/**
[<text position: B25>]
[<line: B20>]
<text inside extension>
/* CIF End TextExtension */

This rule is not used by any other rule, see the additional information.

Additional information:

Left means that the left side of the symbol is open. **Right** means that the right side of the symbol is open. **Right** is default.

If **Left** is given, the symbol and text position defines the upper *right* corner.

The <line: B20> is the line connecting the text extension symbol with the other symbol. If the line is not given, it will be autolayouted. The first point in the pointlist is on the surrounding rectangle of the text extension symbol. The last point in the pointlist is on the surrounding rectangle of the symbol the text extension symbol is attached to.

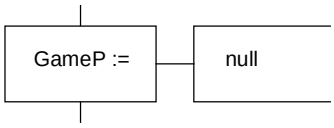
A newline character before or after one of the two CIF text extension comments should not be considered as a part of the text in the symbols.

<Text extension: A25> and <Comment: A26> should be placed before the <end> in the following way: Text extensions and comments may be attached to any rule in the range <block symbol: A30> - <select symbol: A69>.

Example 1 (an informal example with a task symbol):

```
/* CIF Task (800,550) */
TASK 'first part of task text that will be in the task symbol'
/* CIF TextExtension (1100,550) */
'last part of task text that will be in the TextExtension symbol'
/* CIF End TextExtension */
;
```

Example 2:



T1008940-96/d12

```
/* CIF Task (800,550) */
TASK GameP :=
/* CIF TextExtension (1100,550) */
/* CIF Line (1100,600),(1000,600) */
null
/* CIF End TextExtension */
;
```

A26 comment:

/* CIF Comment <position and size: B24> [**Left** | **Right**] [**Dashed**] ***/**
[<text position: B25>]
[<dashed line: B21>]
<comment> <end>

This rule is not used by any other rule, see additional information.

<comment> is
COMMENT <character string>

Additional information:

Left means that the left side of the symbol is open. **Right** means that the right side of the symbol is open. **Right** is default.

If **Left** is given, the symbol (and text) position defines the upper *right* corner.

If **Dashed** is given, the comment symbol should be drawn dashed (as a <comment symbol2> in SDL96). If **Dashed** is not given, the comment symbol should be drawn non-dashed (as a <comment symbol> in SDL92).

The <dashed line: B21> is the line connecting the comment symbol with the other symbol. If the line is not given, it will be autolayouted. The first point in the pointlist is on the surrounding rectangle of the comment symbol. The last point in the pointlist is on the surrounding rectangle of the symbol the comment is attached to.

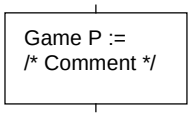
How this CIF construct is used is explained in <text extension: A25>.

Example 1:

T1008950-96/d13

```
/* CIF Task (800,550) */
task GameP:=null
/* CIF Comment (1100,550) */
/* CIF Line (1100,600),(1000,600) Dashed */
COMMENT 'My comment in
a comment symbol';
```

Related example:



T1008950-96/d14

```
/* CIF Task (800,550) */
task GameP:=null
/* Comment */;
```

Example 2:

```
/* CIF Task (800,550) */
task GameP :=
/* CIF TextExtension (1100,550) */
/* CIF Line (1100,600),(1000,600) */
null
/* CIF End TextExtension */
/* CIF Comment (1100,750) */
/* CIF Line (1100,800),(1000,600) Dashed */
COMMENT 'My comment in a comment symbol';
```

A27 create line:

/* CIF CreateLine <pointlist: B22> */

This rule is used by <CIF descriptor: A18>.

Additional information:

The first point in the pointlist is on the surrounding rectangle of the process symbol that creates the other process. The last point in the pointlist is on the surrounding rectangle of the process symbol that is created.

This comment may be placed before or after any symbol or line CIF & PR construct within a diagram as if it was an <entity in block>.

Example:

```
/* CIF DefaultSize (200,100) */
/* CIF Process (200,500) */
PROCESS Main(1,1) REFERENCED;
/* CIF Process (500,500) */
PROCESS Game(0,1) REFERENCED;
/* CIF CreateLine (400,550),(500,550) */
```

A28 flow line:

[<line: B20>]

This rule is used by <CIF descriptor: A18>.

Additional information:

This CIF comment is optional. If the CIF comment is not given for a flowline, the flow line is autolayouted.

The first point in the pointlist is on the surrounding rectangle of the symbol the flow is coming from. The last point in the pointlist is on the surrounding rectangle of the symbol the flow is going to.

This comment may be placed before or after any symbol or line CIF & PR construct within the <process body> of the diagram.

A flowline joining another flowline should describe the complete pointlist from a point on the surrounding rectangle of the "from" symbol to a point on the surrounding rectangle of the "to" symbol.

Arrows on flowlines are implicit. Flowlines after decision and transition option symbols should use <answer flow line: A29>, a CIF rule that is not optional.

Example:

```
/* CIF Start (300,100) */
START;
/* CIF Line (400,200),(400,250) */
/* CIF Set (300,250) */
Set(Now+1,T);
```

A29 answer flow line:

/* CIF Answer [{ **Right** | **Left** }] [**InvisibleBrackets**] */
[<line: B20>] [<text position: B25>]
{ <answer part> | <else part> }

This rule is used by <CIF descriptor: A18>.

<answer part> is
([<answer>] **) :** [<transition>]

<answer> is
<range condition> | <informal text>

<else part> is
ELSE : [<transition>]

Additional information:

This CIF comment should be used for flow lines after a decision or a transition option symbol.

If <line: B20> is given:

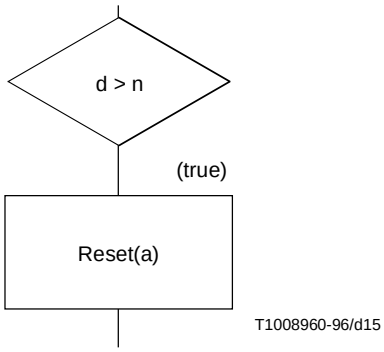
Right and **Left** have no meaning. The pointlist in the flow line specifies where the flow line starts on the decision symbol. The same rules as for <flow line: A28> apply.

If <line: B20> is not given:

Right means that the flow line starts to the right of the decision symbol (or in the lower right corner of the transition option symbol). **Left** means that the flow line starts to the left of the decision symbol (or in the lower left corner of the transition option symbol). As default, the flow line starts below the decision symbol (or in the centre of the bottom edge of the transition option symbol). The rest of the flow line is autolayouted.

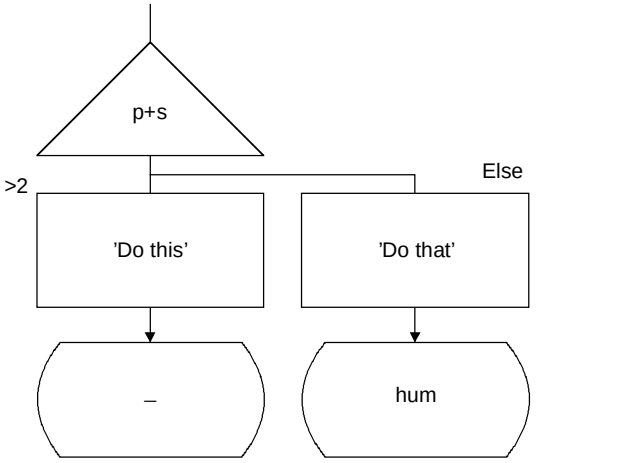
If **InvisibleBrackets** is given, the characters **(** and **)** that can be found in the PR are not visible in GR.

Examples:



T1008960-96/d15

```
       /* CIF Decision (800,550) */
       DECISION d > n;
       /* CIF Answer */
       /* CIF Line (900,650),(900,750) */
       /* CIF TextPosition (910,690) */
       (true):
       /* CIF Reset (800,750) */
       reset(a);
------
```



T1008960-96/d16

```
       /* CIF Alternative (800,550) */
       ALTERNATIVE p + s;
       /* CIF Answer InvisibleBrackets */
       (>2):
       /* CIF Task (800,750) */
       TASK 'Do this';
       /* CIF NextState (800,950) */
       NEXTSTATE -;
       /* CIF Answer */
       ELSE:
       /* CIF Task (1100,750) */
       TASK 'Do that';
       /* CIF NextState (1100,950) */
       NEXTSTATE hum;
```

A30 block symbol:

<block symbol rectangle: B15>
[<text position: B25>]
{ <textual block reference> |
  { <gate reference: B19>* <textual typebased system definition> } }

This rule is used by <CIF descriptor: A18>.

<textual block reference> is
BLOCK <name> **REFERENCED** <end>

<textual typebased block definition> is
BLOCK <typebased block heading> <end>

<typebased block heading> is
<block name> [<number of block instances>] **:** <block type expression>

<number of block instances> is
(<natural simple expression> **)**

Additional information:

<gate reference: B19>s are optional and only used to specify text positions for gate references attached to this block symbol. If a <gate reference: B19> is omitted, the text position for that gate reference will be autolayouted. The name of the gate reference is in PR mentioned in connection with the PR for connected channels or signal routes.

Example 1:

```
/* CIF Block (800,550) */
/* CIF TextPosition (810,560) */
BLOCK myBlock REFERENCED;
```

Example 2:

```
/* CIF Block (800,550) */
/* CIF GateReference (900,550) */
/* CIF TextPosition (890,500) */
BLOCK myBlocks(2):myBlockType;
```

A31 dashed block symbol:

/* CIF Block <position and size: B24> **Dashed** <block name> ***/**
[<text position: B25>]
<gate reference: B19>*

This rule is used by <CIF descriptor: A18>.

Additional information:

This comment may be placed anywhere a <textual block reference> is allowed. There should be one <dashed block symbol: A31> for each <existing typebased block definition> in the SDL-GR. <gate reference: B19> is explained in <block symbol: A30>.

Example:

```
/* CIF Block (800,550) Dashed myBlock */
```

A32 process symbol:

<process symbol rectangle: B16>
[<text position: B25>]
{ <textual process reference> |
  { <gate reference: B19>* <textual typebased process definition> } }

This rule is used by <CIF descriptor: A18>.

<textual process reference> is
PROCESS <name> [<number of process instances>] **REFERENCED** <end>

<textual typebased process definition> is
PROCESS <typebased process heading> <end>

<typebased process heading> is
<process name> [<number of process instances>] **:** <process type expression>

Additional information:

<gate reference: B19> is explained in <block symbol: A30>.

Example 1:

```
/* CIF Process (800,550) */
PROCESS myProcess REFERENCED;
```

Example 2:

```
/* CIF Process (800,550) */
PROCESS myProcess (1,1):myProcessType;
```

A33 dashed process symbol:

/* CIF Process <position and size: B24> **Dashed** <process name> ***/**
[<text position: B25>]
<gate reference: B19>*

This rule is used by <CIF descriptor: A18>.

Additional information:

This comment may be placed anywhere a <textual process reference> is allowed. There should be one <dashed process symbol: A33> for each <existing typebased process definition> in the SDL-GR. <gate reference: B19> is explained in <block symbol: A30>.

Example:

```
/* CIF Process (800,550) Dashed myProcess */
```

A34 service symbol:

<service symbol rectangle: B17>
[<text position: B25>]
{ <textual service reference> |
  { <gate reference: B19>* <textual typebased service definition> } }

This rule is used by <CIF descriptor: A18>.

<textual service reference> is
SERVICE <name> **REFERENCED** <end>

<textual typebased service definition> is
SERVICE <typebased service heading> <end>

<typebased service heading> is
<service name> **:** <service type expression>

Additional information:

<gate reference: B19> is explained in <block symbol: A30>.

Example 1:

```
/* CIF Service (800,550) */
SERVICE myService REFERENCED;
```

Example 2:

```
/* CIF Service (800,550) */
SERVICE myService:myServiceType;
```

A35 dashed service symbol:

/* CIF Service <position and size: B24> **Dashed** <service name> ***/**
[<text position: B25>]
<gate reference: B19>*

This rule is used by <CIF descriptor: A18>.

Additional information:

This comment may be placed anywhere a <textual service reference> is allowed. There should be one <dashed service symbol: A35> for each <existing typebased service definition> in the SDL-GR. <gate reference: B19> is explained in <block symbol: A30>.

Example:

```
/* CIF Service (800,550) Dashed myService */
```

A36 system type symbol:

/* CIF SystemType <position and size: B24> ***/**
[<text position: B25>]
<textual system type reference>

This rule is used by <CIF descriptor: A18>.

<textual system type reference> is
SYSTEM TYPE <name> **REFERENCED** <end>

Example:

```
/* CIF SystemType (800,550) */
SYSTEM TYPE mySystemType REFERENCED;
```

A37 block type symbol:

/* CIF BlockType <position and size: B24> ***/**
[<text position: B25>]
<textual block type reference>

This rule is used by <CIF descriptor: A18>.

<textual block type reference> is
[<virtuality>] **BLOCK TYPE** <name> **REFERENCED** <end>

Example:

```
/* CIF BlockType (800,550) */
VIRTUAL BLOCK TYPE myBlockType REFERENCED;
```

A38 process type symbol:

/* CIF ProcessType <position and size: B24> ***/**
[<text position: B25>]
<textual process type reference>

This rule is used by <CIF descriptor: A18>.

<textual process type reference> is
[<virtuality>] **PROCESS TYPE** <name> **REFERENCED** <end>

Example:

```
/* CIF ProcessType (800,550) */
PROCESS TYPE myProcessType REFERENCED;
```

A39 service type symbol:

/* CIF ServiceType <position and size: B24> ***/**
[<text position: B25>]
<textual service type reference>

This rule is used by <CIF descriptor: A18>.

<textual service type reference> is
[<virtuality>] **SERVICE TYPE** <name> **REFERENCED** <end>

Example:

```
/* CIF ServiceType (800,550) */
SERVICE TYPE myServiceType REFERENCED;
```

A40 block substructure symbol:

/* CIF BlockSubstructure <position and size: B24> ***/**
[<text position: B25>]
<textual block substructure reference>

This rule is used by <CIF descriptor: A18>.

<textual block substructure reference> is
SUBSTRUCTURE <name> **REFERENCED** <end>

Example:

```
/* CIF BlockSubstructure (800,550) */
SUBSTRUCTURE mySubstructure REFERENCED;
```

A41 procedure symbol:

/* CIF Procedure <position and size: B24> ***/**
[<text position: B25>]
<textual procedure reference>

This rule is used by <CIF descriptor: A18>.

<textual procedure reference> is
<procedure preamble> **PROCEDURE** <name> **REFERENCED** <end>

<procedure preamble> is
[<virtuality>] [**EXPORTED** [**AS** <remote procedure identifier>]]

Example:

```
/* CIF Procedure (800,550) */
VIRTUAL PROCEDURE myProcedure REFERENCED;
```

A42 operator symbol:

/* CIF Operator <name> <position and size: B24> ***/**
[<text position: B25>]

This rule is used by <CIF descriptor: A18>.

Additional information:

This comment may be placed anywhere an <entity in package>, <entity in system>, <entity in block>, <entity in process>, <entity in service> or an <entity in procedure> may be placed. The operator symbol has no direct graphical connection to the <textual operator reference> (that is presented as text in a text symbol).

Note that this rule is a compromise because the operator GR symbol is not standardized. To be complete, the operator should not just be identified by name. Qualifier, parameters and result have to be considered also. Another complication is that the name could be replaced by a <quoted operator>.

Example:

```
/* CIF Operator myOperator (800,550) */
```

A43 start symbol:

/* CIF Start <position and size: B24> ***/**
[<text position: B25>]
<beginning of start>

This rule is used by <CIF descriptor: A18>.

<beginning of start> is
START [<virtuality>] <end>

Additional information:

This CIF comment should be used in process and process type diagrams. Procedure and operator diagrams should use <procedure start symbol: A54>.

Example:

```
/* CIF Start (800,550) */
START;
```

A44 stop symbol:

/* CIF Stop <position and size: B24> ***/**
<stop> <end>

This rule is used by <CIF descriptor: A18>.

<stop> is
STOP

Example:

```
/* CIF Stop (800,550) */
STOP;
```

A45 state symbol:

/* CIF State <position and size: B24> ***/**
[<text position: B25>]
<beginning of state>

This rule is used by <CIF descriptor: A18>.

<beginning of state> is
STATE <state list> <end>

<state list> is
{ <state name> { **,** <state name> }* } | <asterisk state list>

Examples for this CIF comment can be found in the related CIF comment <nextstate symbol: A46>.

A46 nextstate symbol:

/* CIF NextState <position and size: B24> ***/**
[<text position: B25>]
<nextstate> <end>

This rule is used by <CIF descriptor: A18>.

<nextstate> is
NEXTSTATE <nextstate body>

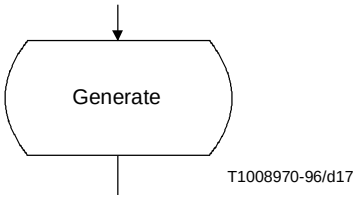
<nextstate body> is
{ <state name> | <dash nextstate> }

<dash nextstate> is
<hyphen>

Additional information:

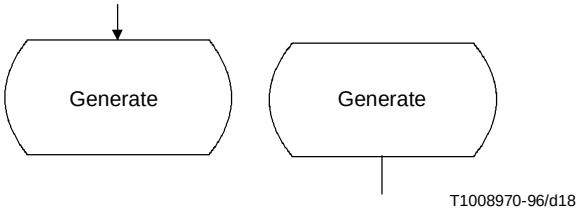
A CIF comment should be given for every state and every nextstate in PR. This means that there will be two CIF comments for one GR symbol that corresponds to both a PR state and a PR nextstate. A tool that reads a CIF file should determine if a state and a nextstate is in fact one GR symbol by comparing the co-ordinates of the symbols in the two CIF comments.

Examples:



T1008970-96/d17

```
/* CIF NextState (800,550) */
NEXTSTATE Generate;
/* CIF State (800,550) */
STATE Generate;
```



T1008970-96/d18

```
/* CIF NextState (800,550) */
NEXTSTATE Generate;
/* CIF State (1100,550) */
STATE Generate;
```

A47 save symbol:

/* CIF Save <position and size: B24> ***/**
[<text position: B25>]
<basic save part>

This rule is used by <CIF descriptor: A18>.

<basic save part> is (SDL92)
(…the same as <save part> in SDL96. Both <basic save part> in SDL92 and <save part> in SDL96 should use this CIF comment.)
SAVE [<virtuality>] <save list> <end>

<save list> is
{ <signal list> | <asterisk save list> }

Example:

```
/* CIF Save (800,550) */
SAVE mySignal;
```

A48 task symbol:

/* CIF Task <position and size: B24> ***/**
[<text position: B25>]
<task> <end>

This rule is used by <CIF descriptor: A18>.

<task> is
TASK <task body>

<task body> is
{ <assignment statement> { **,** <assignment statement> }* } |
{ <informal text> { **,** <informal text> }* }

<informal text> is
<character string>

<character string> is
<apostrophe> { <alphanumeric> | <other character> | <special> | <full stop> | <underline> | <space> | {<apostrophe><apostrophe>} }* <apostrophe>

Additional information:

There are three cases when a task symbol should be described by other CIF comments than this one: GR task symbols containing <set> should use <set symbol: A49>. GR task symbols containing <reset> should use <reset symbol: A50>. GR task symbols containing <export> should use <export symbol: A51>.

Example:

```
/* CIF Task (800,550) */
TASK myVariable := 0;
```

A49 set symbol:

/* CIF Set <position and size: B24> ***/**
[<text position: B25>]
<set> <end>

This rule is used by <CIF descriptor: A18>.

<set> is
SET <set statement> { **,** <set statement> }*

<set statement> is
([<time expression> **,**] <timer identifier> [**(** <expression list> **)**] **)**

<expression list> is
<expression> { **,** <expression> }*

Additional information:

A set symbol is a GR task symbol containing <set>.

Example:

```
/* CIF Set (800,550) */
SET (Now+1, myTime);
```

A50 reset symbol:

/* CIF Reset <position and size: B24> ***/**
[<text position: B25>]
<reset> <end>

This rule is used by <CIF descriptor: A18>.

<reset> is
RESET <reset statement> { **,** <reset statement> }*

<reset statement> is
<timer identifier> [<expression list>]

Additional information:

A reset symbol is a GR task symbol containing <reset>.

Example:

```
/* CIF Reset (800,550) */
RESET T;
```

A51 export symbol:

/* CIF Export <position and size: B24> ***/**
[<text position: B25>]
<export> <end>

This rule is used by <CIF descriptor: A18>.

<export> is
EXPORT (<variable identifier> { **,** <variable identifier> }* **)**

Additional information:

An export symbol is a GR task symbol containing <export>.

Example:

```
/* CIF Export (800,550) */
Export (myVariable1, myVariable2);
```

A52 create symbol:

/* CIF Create <position and size: B24> ***/**
[<text position: B25>]
<create request> <end>

This rule is used by <CIF descriptor: A18>.

<create request> is
CREATE <create body>

<create body> is
{ <process identifier> | **THIS** } [<actual parameters>]

<actual parameters> is
([<expression>] { **,** [<expression>] }* **)**

Example:

```
/* CIF Create (800,550) */
CREATE Game;
```

A53 procedure call symbol:

/* CIF ProcedureCall <position and size: B24> ***/**
[<text position: B25>]
<procedure call> <end>

This rule is used by <CIF descriptor: A18>.

<procedure call> is
CALL <procedure call body>

<procedure call body> is
[**THIS**] <procedure identifier> [<actual parameters>]

Example:

```
/* CIF ProcedureCall (800,550) */
CALL myProcedure;
```

A54 procedure start symbol:

/* CIF ProcedureStart <position and size: B24> ***/**
[<text position: B25>]
<beginning of start>

This rule is used by <CIF descriptor: A18>.

<beginning of start> is
START [<virtuality>] <end>

Additional information:

This CIF comment should be used for procedure and operator diagrams. Process and process type diagrams should use <start symbol: A43>.

Example:

```
/* CIF ProcedureStart (800,550) */
START;
```

A55 return symbol:

/* CIF Return <position and size: B24> ***/**
[<text position: B25>]
<return> <end>

This rule is used by <CIF descriptor: A18>.

<return> is
RETURN [<expression>]

Example:

```
/* CIF Return (800,550) */
RETURN myReturnValue;
```

A56 decision symbol:

/* CIF Decision <position and size: B24> ***/**
[<text position: B25>]
<beginning of decision>

This rule is used by <CIF descriptor: A18>.

<beginning of decision> is
DECISION <question> <end>

<question> is <question expression> | <informal text> | **ANY**

Additional information:

A flowline directly after a decision symbol should be described by an <answer flow line: A29>.

Example (without flow lines):

```
/* CIF Decision (800,550) */
DECISION DoorIndex > NoOfDoors;
```

A57 continuous signal symbol:

/* CIF ContinuousSignal <position and size: B24> ***/**
[<text position: B25>]
<beginning of continuous signal>

This rule is used by <CIF descriptor: A18>.

<beginning of continuous signal> is
PROVIDED [<virtuality>] <boolean expression> <end> [**PRIORITY** <integer literal name> <end>]

Example:

```
/* CIF ContinuousSignal (800,550) */
PROVIDED level > 5;
```

A58 enabling condition symbol:

/* CIF EnablingCondition <position and size: B24> ***/**
[<text position: B25>]
<enabling condition>

This rule is used by <CIF descriptor: A18>.

<enabling condition> is
PROVIDED <boolean expression> <end>

Example:

```
/* CIF EnablingCondition (800,550) */
PROVIDED level > 5;
```

A59 transition option symbol:

/* CIF TransitionOption <position and size: B24> ***/**
[<text position: B25>]
<beginning of transition option>

This rule is used by <CIF descriptor: A18>.

<beginning of transition option> is
ALTERNATIVE <alternative question> <end>

<alternative question> is
<simple expression> | <informal text>

Additional information:

A flowline directly after a transition option symbol should be described by an <answer flow line: A29>.

Example:

```
/* CIF TransitionOption (800,550) */
ALTERNATIVE level;
```

A60 join symbol:

/* CIF Join { <position and size: B24> | **Invisible** } ***/**
[<text position: B25>]
<join> <end>

This rule is used by <CIF descriptor: A18>.

<join> is
JOIN <connector name>

Additional information:

Invisible means that this PR join should not be visible as a symbol in GR, it is only given in PR to indicate a flowline ending in an already described symbol (i.e. the symbol following the label <join> is referring to). See also <label symbol: A64>.

Example 1:

```
/* CIF Join Invisible */
JOIN myInvisibleLabel;
```

Example 2:

```
/* CIF Join (800,550) */
JOIN myLabel;
```

A61 macro call symbol:

/* CIF MacroCall <position and size: B24> ***/**
[<text position: B25>]
[<inlet text: B10>]
[<outlet text: B11>]
<macro call>

This rule is used by <CIF descriptor: A18>.

<macro call> is
MACRO <macro name> [<macro call body>] <end>

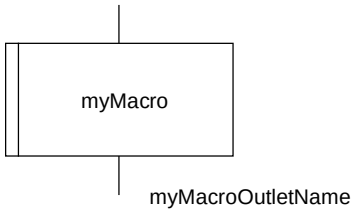
<macro call body> is
(<macro actual parameter> { **,** <macro actual parameter> }* **)**

<macro actual parameter> is
{ <lexical unit> }*

Additional information:

<inlet text: B10> should be given if the macro call symbol is associated with a macro inlet label. <outlet text: B11> should be given if the macro call symbol is associated with a macro outlet label. Read more about macros in <macro diagram start: A16>.

Example:



T1008980-96/d19

```
/* CIF MacroCall (800,550) */
/* CIF OutletText myMacroOutletName */
MACRO myMacro;
```

A62 macro inlet symbol:

/* CIF MacroInlet <position and size: B24> ***/**
[<inlet text: B10>]
<macro body>

This rule is used by <CIF descriptor: A18>.

<macro body> is
{ <lexical unit> | <formal name> }*

Additional information:

<inlet text: B10> should be given if the macro inlet symbol is associated with a macro label. Read more about macros in <macro diagram start: A16>.

Example (on a macro inlet symbol without a label followed by an output symbol):

```
/* CIF MacroInlet (800,550) */
/* CIF Output (800,750) */
OUTPUT outSignal;
```

A63 macro outlet symbol:

/* CIF MacroOutlet <position and size: B24> ***/**
[<outlet text: B11>]
<diagram end: A17>
<end of macro definition>

This rule is used by <CIF descriptor: A18>.

<end of macro definition> is
ENDMACRO [<macro name>] <end>

Additional information:

<outlet text: B11> should be given if the macro outlet symbol is associated with a macro label. Read more about macros in <macro diagram start: A16>.

Example:

```
/* CIF MacroOutlet (800,950) */
/* CIF End MacroDiagram */
ENDMACRO myMacro;
```

A64 label symbol:

/* CIF Label { <position and size: B24> | **Invisible** } ***/**
[<text position: B25>]
(<label> | <beginning of free action>)

This rule is used by <CIF descriptor: A18>.

<label> is
<connector name> :

<beginning of free action> is
CONNECTION <beginning of transition>

Additional information:

Invisible means that this PR label should not be visible as a symbol in GR, it is only given in PR to indicate a flowline ending in an already described symbol (i.e. the symbol following the label). See also <join symbol: A60>.

The first <label> in a <beginning of free action> has its CIF comment located before <beginning of free action>, see the last example below.

Example 1:

```
/* CIF Label Invisible */
myInvisibleLabel:
```

Example 2:

```
/* CIF Label (800,550),(100,100) */
myVisibleLabel:
```

Example 3:

```
/* CIF Label (800,550), (100,100) */
CONNECTION myLabel:
```

A65 input symbol:

/* CIF Input <position and size: B24> [{ **Left** | **Right** }] ***/**
[<text position: B25>]
{ <beginning of basic input part> | <beginning of spontaneous transition> | <beginning of remote procedure input transition> }

This rule is used by <CIF descriptor: A18>.

<beginning of basic input part> is (SDL92)
(…the same as <beginning of input part> in SDL96. Both <beginning of basic input part> in SDL92 and <beginning of input part> in SDL96 should use this CIF comment.)
INPUT [<virtuality>] <input list> <end>

<input list> is
<asterisk> | { <stimulus> { **,** <stimulus> }* }

<stimulus> is (SDL92)
{ <signal identifier> | <timer identifier> } [**(** [<variable>] { **,** [<variable>] }* **)**]

<stimulus> is (SDL96)
<signal list item> [**(** [<variable>] { **,** [<variable>] }* **)**]

<beginning of spontaneous transition> is
INPUT [<virtuality>] **NONE** <end>

<beginning of remote procedure input transition> is (SDL92)
INPUT [<virtuality>] **PROCEDURE** <remote procedure identifier list> <end>

<remote procedure identifier list> is
<remote procedure identifier> { **,** <remote procedure identifier> }*

Additional information:

Left means that the part of the symbol that visualizes an arrow is to the left. **Right** means that the part of the symbol that visualizes an arrow is to the right. Default is **Right**.

Example:

```
/* CIF Input (800,550) Left */
INPUT mySignal;
```

A66 priority input symbol:

/* CIF PriorityInput <position and size: B24>
[{ **Left** | **Right** }] ***/**
[<text position: B25>]
<beginning of priority input>

This rule is used by <CIF descriptor: A18>.

<beginning of priority input> is
PRIORITY INPUT [<virtuality>] <priority input list> <end>

<priority input list> is
<stimulus> { **,** <stimulus> }*

Additional information:

Left means that the part of the symbol that visualizes an arrow is to the left. **Right** means that the part of the symbol that visualizes an arrow is to the right. Default is **Right**.

Example:

```
/* CIF PriorityInput (800,550) Left */
PRIORITY INPUT mySignal;
```

A67 output symbol:

/* CIF Output <position and size: B24> [{ **Left** | **Right** }] ***/**
[<text position: B25>]
<output> <end>

This rule is used by <CIF descriptor: A18>.

<output> is
OUTPUT <output body>

<output body> ::=
<signal identifier> [<actual parameters>] { **,** <signal identifier> [<actual parameters>] }* [**TO** <destination>] [**VIA** [**ALL**] <via path>]

<destination> is
<PId expression> | <process identifier> | **THIS**

<via path> is
<via path element> { **,** <via path element> }*

<via path element> is
<signal route identifier> | <channel identifier> | <gate identifier>

Additional information:

Left means that the part of the symbol that visualizes an arrow is to the left. **Right** means that the part of the symbol that visualizes an arrow is to the right. Default is **Right**.

Example:

```
/* CIF Output (800,550) */
OUTPUT mySignal;
```

A68 text symbol:

/* CIF Text <position and size: B24> ***/**
[<text position: B25>]
<text in text symbol>
/* CIF End Text */

This rule is used by <CIF descriptor: A18>.

<text in text symbol> is
self explanatory (and not a PR construct).

Additional information:

A newline character before or after one of the two CIF text comments should not be considered as a part of the text in the text symbol.

Example:

```
/* CIF Text (800,550) */
Timer myTimer;
/* CIF End Text */
```

A69 select symbol:

/* CIF Select <pointlist: B22> ***/**
[<text position: B25>]
<beginning of select definition>

This rule is used by <CIF descriptor: A18>.

<beginning of select definition> is
SELECT IF <boolean simple expression> <end>

Additional information:

The points in the pointlist describe all the corners of the select symbol in either clockwise or counter clockwise order.

Example:

```
/* CIF Select (700,400),(1100,400),(1100,750),(700,750) */
/* CIF TextPosition (725,425) */
SELECT IF (p = 3);
```

A70 descriptor end:

{ /* **CIF End Decision** */ <end of decision> |
/* **CIF End State** */ <part of end of state> |
/* **CIF End Label** */ <part of end of free action> |
/* **CIF End Select** */ <end of select definition> |
/* **CIF End TransitionOption** */ <end of transition option> **}**

This rule is used by <CIF descriptor: A18>.

<end of decision> is
ENDDECISION

<part of end of state> is
ENDSTATE [<state name>] <end>

<part of end of free action> is
ENDCONNECTION [<connector name>] <end>

<end of select definition> is
ENDSELECT <end>

<end of transition option> is
ENDALTERNATIVE

Additional information:

This rule is introduced to distinguish end of information about a symbol from end of information about a diagram.

6.4.2 CIF B rules

B1 diagram parts:

{ <page declaration: B3>+ | <nested frame: B13> }

This rule is used by <system type diagram start: A6>, <block diagram start: A7>, <process diagram start: A10>, <service diagram start: A12>, <service type diagram start: A13>, <procedure diagram start: A14>, <operator diagram start: A15>.

Additional information:

If the diagram is embedded in the enclosing diagram, a <nested frame: B13> should be used.

If the diagram is not embedded in the enclosing diagram, there should be one <page declaration: B3> for each page in the diagram.

B2 diagram parts & gate references:

{ <page declaration: B3> [<gate reference: B19>+] }+ |
{ <nested frame: B13> [<gate reference: B19>+] }

This rule is used by <block type diagram start: A8>, <substructure diagram start: A9>, <process type diagram start: A11>.

Additional information:

If the diagram is embedded in the enclosing diagram, a <nested frame: B13> should be used.

If the diagram is not embedded in the enclosing diagram, there should be one <page declaration: B3> for each page in the diagram.

<gate reference: B19>s are optional. They are only used to specify text positions for gate references attached to the diagram frame on the previously defined page/nested frame. If a <gate reference: B19> is omitted, the text position will be autolayouted. The name of a gate reference is mentioned in PR in connection with the PR for connected signal routes or channels.

<gate reference: B19>s do not exist on ServicePages.

B3 page declaration:

/* CIF <page type: B12> <page name> <size point: B26> ***/**
[<frame declaration: B14>]
[<diagram heading text position: B25>]
[<page text position: B23>]
[<package reference: B18>]

This rule is used by <package diagram start: A4>, <system diagram start: A5>, <macro diagram start: A16>, <diagram parts: B1>, <diagram parts & gate references: B2>.

Additional information:

If a <frame declaration: B14> is not given, the frame has the position (0,0) and the same size as the page. This page becomes the current page and remains the current page until either a new <page declaration: B3> or a <page switch: A20> is encountered.

The <page text position: B23> defines the upper *right* corner of the surrounding rectangle of the text.

A <package reference: B18> may only exist if the diagram is a package or system diagram.

The <package reference: B18> should only be given if the package reference symbol is visible in SDL-GR.

Example:

```
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,250),(1700,1950) */
```

B4 channel substructure symbol:

/* CIF ChannelSubstructure <position and size: B24> ***/**
[<text position: B25>]
<dashed line: B21>

This rule is used by <channel: A21>.

Additional information:

The pointlist describes the <dashed association symbol> that connects the channel substructure symbol and the channel. The first point in the pointlist is on the surrounding rectangle of the channel substructure symbol. The last point in the pointlist is on the line defined by the pointlist for the channel the channel substructure is connected to.

B5 gate constraint symbol:

<block symbol rectangle: B15> | <process symbol rectangle: B16> | <service symbol rectangle: B17>

This rule is used by <gate: A23>.

B6 first signallist text position:

/* CIF TextPosition <point: B26> **SignalList1 */**

This rule is used by <channel: A21>, <signal route: A22>, <gate: A23>.

Additional information:

This text position should be tied to the first signal list mentioned in the PR code.

Example:

```
/* CIF TextPosition (800,550) SignalList1 */
```

B7 second signallist text position:

/* CIF TextPosition <point: B26> **SignalList2 */**

This rule is used by <channel: A21>, <signal route: A22>, <gate: A23>.

Additional information:

This text position should be tied to the second signal list mentioned in the PR code.

B8 first arrow position:

/* CIF Arrow1Position <point: B26> ***/**

This rule is used by <channel: A21>.

Additional information:

This arrow position is used for the first channel path mentioned in the PR for the <channel: A21>.

Example:

```
/* CIF Arrow1Position (800,550) */
```

B9 second arrow position:

/* CIF Arrow2Position <point: B26> ***/**

This rule is used by <channel: A21>.

Additional information:

This arrow position is used for the second channel path mentioned in the PR for the <channel: A21>.

B10 inlet text:

/* CIF InletText <macro label> ***/**
[<text position: B25>]

This rule is used by <macro call symbol: A61>, <macro inlet symbol: A62>.

B11 outlet text:

/* CIF OutletText <macro label> ***/**
[<text position: B25>]

This rule is used by <macro call symbol: A61>, <macro outlet symbol: A63>.

B12 page type:

Page | **BlockPage** | **ServicePage**

This rule is used by <page declaration: B3>.

Additional information:

In the normal case **Page** should be used.

BlockPage is used in block and block type diagrams to distinguish between pages containing blocks and pages containing processes. If **BlockPage** is given, the page contains blocks. If **Page** is given, the page contains processes.

ServicePage is used in process and process type diagrams to distinguish between pages containing flow graphs and pages containing services. If **ServicePage** is given, the page contains services. If **Page** is given, the page contains flow graphs. Note that all pages in a process diagram must be of the same type.

B13 nested frame:

/* CIF NestedFrame <position and size: B24> ***/**
[<diagram heading text position: B25>]

This rule is used by <diagram parts: B1>, <diagram parts & gate references: B2>.

B14 frame declaration:

/* CIF Frame <position and size: B24> ***/**

This rule is used by <page declaration: B3>.

B15 block symbol rectangle:

/* CIF Block <position and size: B24> ***/**

This rule is used by <block symbol: A30>, <gate constraint symbol: B5>.

B16 process symbol rectangle:

/* CIF Process <position and size: B24> ***/**

This rule is used by <process symbol: A32>, <gate constraint symbol: B5>.

B17 service symbol rectangle:

/* CIF Service <position and size: B24> ***/**

This rule is used by <service symbol: A34>, <gate constraint symbol: B5>.

B18 package reference:

/* CIF PackageReference <position and size: B24> ***/**
[<text position: B25>]

This rule is used by <page declaration: B3>.

B19 gate reference:

/* CIF GateReference <connection point: B26> ***/**
<text position: B25>

This rule is used by <block symbol: A30>, <dashed block symbol: A31>, <process symbol: A32>, <dashed process symbol: A33>, <service symbol: A34>, <dashed service symbol: A35>, <diagram parts & gate references: B2>.

Additional information:

This CIF comment is used to specify the text position for a gate reference. The connection point is the point where the gate reference connects to channels or signal routes. The name of the gate can be found in the PR for connected channels or signal routes. Read more about gates and gate references in <channel: A21>. See also the example in the examples document. Use this CIF comment for connections via gates. Use <connect: A24> for direct connections between channels and signal routes without gates.

Example:

```
/* CIF GateReference (800,550) */
/* CIF TextPosition (750,500) */
```

B20 line:

/* CIF Line <pointlist: B22> ***/**

This rule is used by <text extension: A25>, <flow line: A28>, <answer flow line: A29>.

B21 dashed line:

/* CIF Line <pointlist: B22> **Dashed */**

This rule is used by <comment: A26>, <channel substructure symbol: B4>.

B22 pointlist:

<first point: B26> { **,** <next point: B26> }+

This rule is used by <channel: A21>, <signal route: A22>, <gate: A23>, <create line: A27>, <select symbol: A69>, <line: B20>, <dashed line: B21>.

B23 page text position:

/* CIF TextPosition <point: B26> **PageName */**

This rule is used by <page declaration: B3>.

B24 position and size:

<position point: B26> [**,** <size point: B26>]

This rule is used by <text extension: A25>, <comment: A26>, <dashed block symbol: A31>, <dashed process symbol: A33>, <dashed service symbol: A35>, <system type symbol: A36>, <block type symbol: A37>, <process type symbol: A38>, <service type symbol: A39>, <block substructure symbol: A40>, <procedure symbol: A41>, <operator symbol: A42>, <start symbol: A43>, <stop symbol: A44>, <state symbol: A45>, <nextstate symbol: A46>, <save symbol: A47>, <task symbol: A48>, <set symbol: A49>, <reset symbol: A50>, <export symbol: A51>, <create symbol: A52>, <procedure call symbol: A53>, <procedure start symbol: A54>, <return symbol: A55>, <decision symbol: A56>, <continuous signal symbol: A57>, <enabling condition symbol: A58>, <transition option symbol: A59>, <join symbol: A60>, <macro call symbol: A61>, <macro inlet symbol: A62>, <macro outlet symbol: A63>, <label symbol: A64>, <input symbol: A65>, <priority input symbol: A66>, <output symbol: A67>, <text symbol: A68>, <nested frame: B13>, <frame declaration: B14>, <channel substructure symbol: B4>, <block symbol rectangle: B15>, <process symbol rectangle: B16>, <service symbol rectangle: B17>, <package reference: B18>.

Additional information:

The size point may be omitted if a default size is defined earlier with <default size: A19>. The position point is the upper *left* corner of the surrounding rectangle if nothing else is specified in a higher rule.

B25 text position:

/* CIF TextPosition <point: B26> ***/**

This rule is used by <channel: A21>, <signal route: A22>, <gate: A23>, <connect: A24>, <text extension: A25>, <comment: A26>, <answer flow line: A29>, <block symbol: A30>, <dashed block symbol: A31>, <process symbol: A32>, <dashed process symbol: A33>, <service symbol: A34>, <dashed service symbol: A35>, <system type symbol: A36>, <block type symbol: A37>, <process type symbol: A38>, <service type symbol: A39>, <block substructure symbol: A40>, <procedure symbol: A41>, <operator symbol: A42>, <start symbol: A43>, <state symbol: A45>, <nextstate symbol: A46>, <save symbol: A47>, <task symbol: A48>, <set symbol: A49>, <reset symbol: A50>, <export symbol: A51>, <create symbol: A52>, <procedure call symbol: A53>, <procedure start symbol: A54>, <return symbol: A55>, <decision symbol: A56>, <continuous signal symbol: A57>, <enabling condition symbol: A58>, <transition option symbol: A59>, <join symbol: A60>, <macro call symbol: A61>, <label symbol: A64>, <input symbol: A65>, <priority input symbol: A66>, <output symbol: A67>, <text symbol: A68>, <select symbol: A69>, <page declaration: B3>, <nested frame: B13>, <channel substructure symbol: B4>, <package reference: B18>, <gate reference: B19>, <inlet text: B10>, <outlet text: B11>.

Additional information:

The point defines the upper *left* corner of the surrounding rectangle of the text if nothing else is specified in a higher rule. The width and height of the text is not defined.

Example:

```
/* CIF TextPosition (800,550) */
```

B26 point:

(<integer> **,** <integer> **)**

This rule is used by <default size: A19>, <page declaration: B3>, <gate reference: B19>, <pointlist: B22>, <text position: B25>, <page text position: B23>, <first signallist text position: B6>, <second signallist text position: B7>, <first arrow position: B8>, <second arrow position: B9>, <position and size: B24>.

6.5 Tool-specific CIF comments

C0 tool-specific CIF comment:

/* CIF [**Keep**] **Specific** <name of tool> <tool-specific information> ***/**

Additional information:

If **Keep** is given, this tool-specific CIF comment should be kept if the current CIF object is edited. If **Keep** is omitted, this CIF comment should be removed when the current CIF object is edited.

A tool-specific CIF comment should be associated with an A rule (equal to CIF object). The tool-specific CIF comment should be placed between the CIF comments and the SDL-PR constructs associated with that A rule. The order between tool-specific CIF comments associated with the same A rule is undefined.

Informal example (not correct CIF and not correct SDL-PR):

```
/* CIF 'The CIF comment associated with rule Ax'*/
/* CIF Specific mySDLTool 'This information is understood by mySDLTool.
Tools other than mySDLTool may or may not understand this information.
This tool-specific CIF comment is associated with rule Ax'*/
'This is the SDL-PR associated with A rule number X';
```

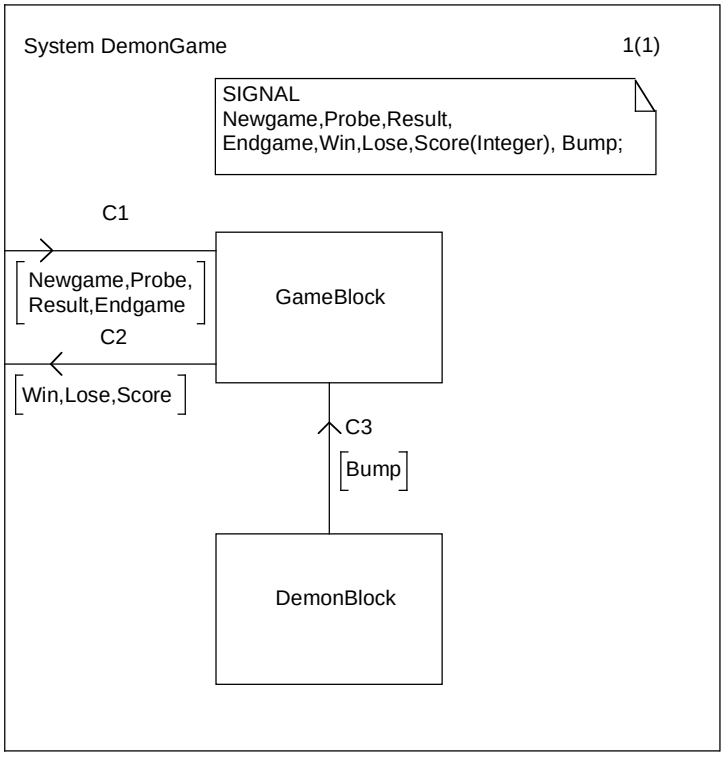
7 Examples

This clause shows some SDL-GR examples and the corresponding SDL-CIF. First three complete (but small) diagrams are presented, then some tricky SDL-GR constructs are presented and discussed.

Note that most of the examples given here are not nested diagrams, i.e. the SDL-PR uses the keyword REFERENCED. It is of course also allowed in CIF to express nested diagrams, where the SDL-PR does not have the keyword REFERENCED.

7.1 System DemonGame

7.1.1 System DemonGame in SDL-GR

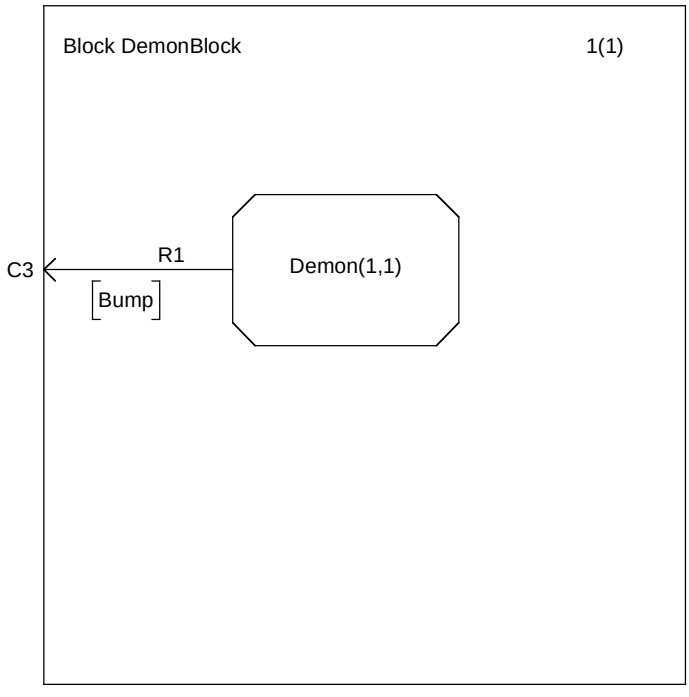


T1008990-96/d20

System DemonGame in SDL-CIF

```
/* CIF SystemDiagram */
/* CIF Page 1 (1150,1000) */
System DemonGame;
/* CIF Specific mySDLTool Page 1 NoGrid */
/* CIF DefaultSize (300,200) */
/* CIF Text (400,100),(600,100) */
SIGNAL
Newgame,Probe,Result,
Endgame,Win,Lose,Score(Integer),Bump;
/* CIF End Text */
/* CIF Channel (550,700),(550,500) */
/* CIF TextPosition (575,551) */
/* CIF TextPosition (575,600) SignalList1 */
channel C3
from DemonBlock to GameBlock
with Bump;
endchannel C3;
/* CIF Channel (400,475),(0,475) */
/* CIF TextPosition (150,425) */
/* CIF TextPosition (62,500) SignalList1 */
channel C2
from GameBlock to env
with Win,Lose,Score;
endchannel C2;
/* CIF Channel (0,325),(400,325) */
/* CIF TextPosition (212,250) */
/* CIF TextPosition (50,350) SignalList1 */
channel C1
from env to GameBlock
with Newgame,Probe,
Result,Endgame;
endchannel C1;
/* CIF BlockSymbol (400,700) */
block DemonBlock referenced;
/* CIF BlockSymbol (400,300) */
block GameBlock referenced;
/* CIF End SystemDiagram */
endsystem DemonGame;
```

7.1.2 Block DemonBlock in SDL-GR

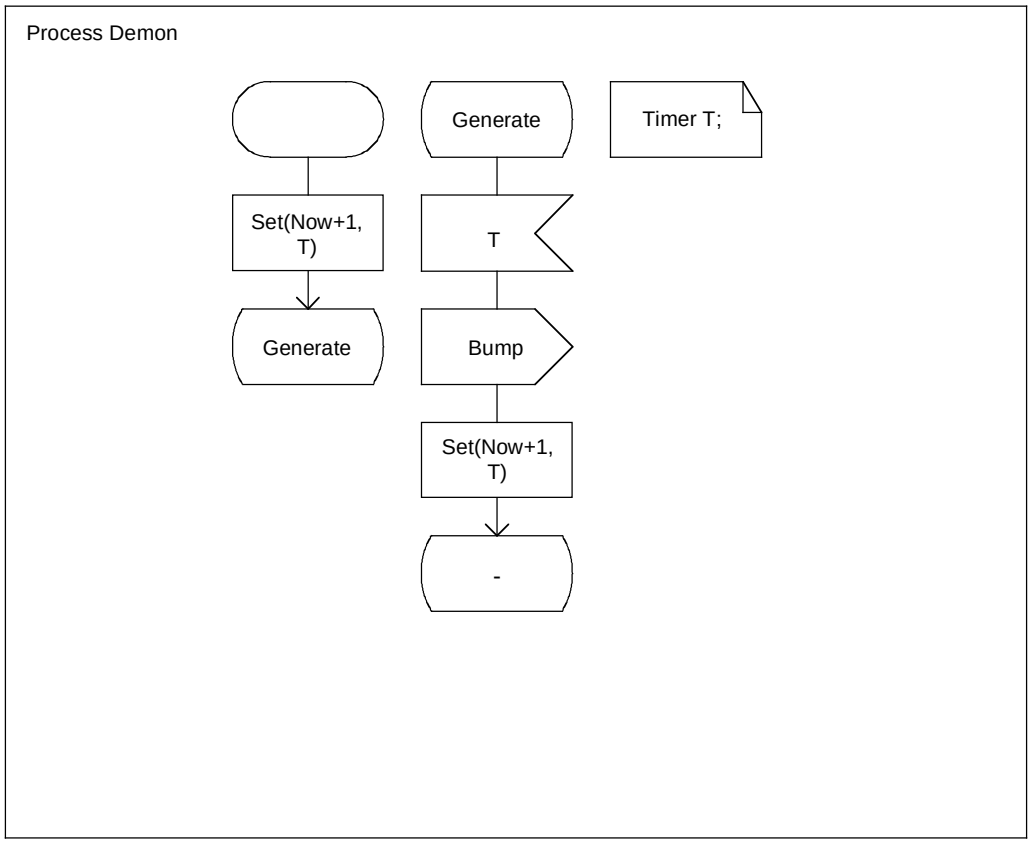


T1009000-96/d21

Block DemonBlock in SDL-CIF

```
/* CIF BlockDiagram */
/* CIF Page 1 (1000,1000) */
/* CIF Frame (100,100),(800,800) */
Block DemonBlock;
/* CIF Specific mySDLTool Page 1 NoGrid */
/* CIF DefaultSize (300,200) */
/* CIF Connect */
/* CIF TextPosition (25,300) */
Connect C3 and R1;
/* CIF SignalRoute (250,350),(0,350) */
/* CIF TextPosition (150,300) */
/* CIF TextPosition (75,375) SignalList1 */
signalroute R1
from Demon to env with Bump;
/* CIF Process (250,250) */
process Demon (1,1) referenced;
/* CIF End BlockDiagram */
endblock DemonBlock;
```

7.1.3 Process Demon in SDL-GR



T1009010-96/d22

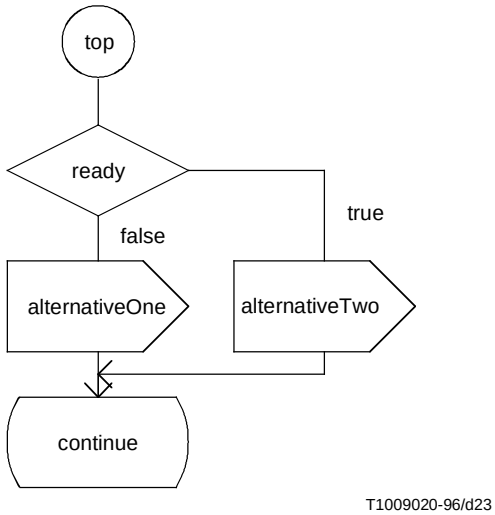
Process Demon in SDL-CIF (without flowlines)

```
/* CIF ProcessDiagram */
/* CIF Page 1 (1400,1000) */
Process Demon ;
/* CIF DefaultSize (200,100) */
/* CIF Text (800,100) */
Timer T;
```

```
/* CIF End Text */
/* CIF Start (300,100) */
start ;
/* CIF Set (300,250) */
Set(Now+1,
T) ;
/* CIF NextState (300,400) */
nextstate Generate ;
/* CIF State (550,100) */
state Generate ;
/* CIF Input (550,250) */
input T ;
/* CIF Output (550,400) */
output Bump ;
/* CIF Set (550,550) */
Set(Now+1,
T) ;
/* CIF NextState (550,700) */
nextstate - ;
/* CIF End ProcessDiagram */
endprocess Demon;
```

7.2 Tricky SDL constructs

7.2.1 Joining flowlines 1



Note that the pointlist for the flowline after the output symbol alternativeTwo has four points, two of them on the border of symbols.

```
/* CIF Label (50,100),(100,100) */
top :
  /* CIF Line (100,200),(100,300) */
  /* CIF Decision (0,300) */
  decision ready;
  /* CIF Answer */
  /* CIF Line (100,400),(100,500) */
  (false) :
    /* CIF Output (0,500) */
    output alternativeOne;
  /* CIF Line (100,600),(100,700) */
  /* CIF Answer */
```

```
/* CIF Line (200,350),(400,350),(400,500) */
(true) :
  /* CIF Output (300,500) */
  output alternativeTwo;
/* CIF Line (400,600),(400,650),(100,650),(100,700) */
/* CIF End Decision */
enddecision;
/* CIF NextState (0,700) */
nextstate continue;
```

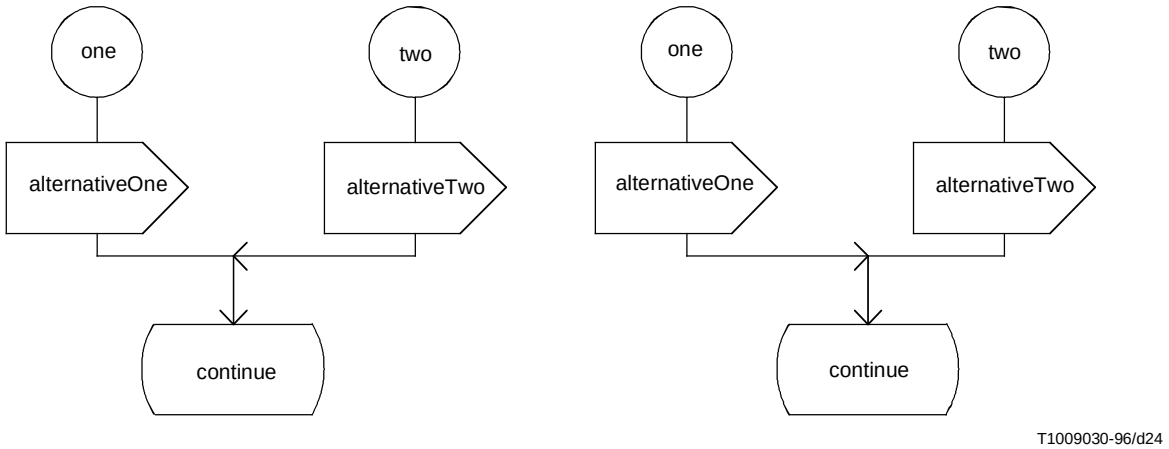
7.2.2 Joining flowlines 2

This example is very similar to the previous one, in fact it is the same GR. The PR is expressed differently though. Note that the two joining flowlines are placed close to the symbol they are coming from in the PR below.

```
/* CIF Label (50,100),(100,100) */
top :
/* CIF Line (100,200),(100,300) */
/* CIF Decision (0,300) */
decision ready;
/* CIF Answer */
/* CIF Line (100,400),(100,500) */
(false) :
  /* CIF Output (0,500) */
  output alternativeOne;
  /* CIF Line (100,600),(100,700) */
  /* CIF Label Invisible */
  grs0 :
  /* CIF NextState (0,700) */
  nextstate continue;
/* CIF Answer */
/* CIF Line (200,350),(400,350),(400,500) */
(true) :
  /* CIF Output (300,500) */
  output alternativeTwo;
  /* CIF Line (400,600),(400,650),(100,650),(100,700) */
  /* CIF Join Invisible */
  join grs0;
/* CIF End Decision */
enddecision;
```

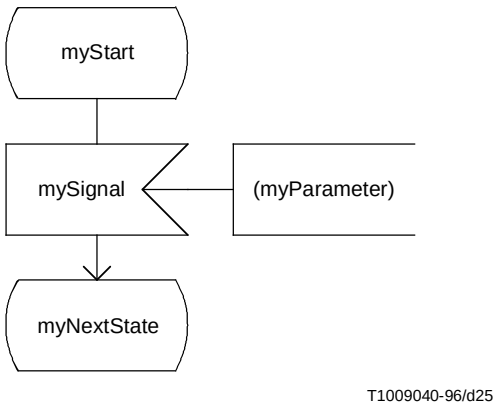
7.2.3 Joining flowlines 3

CIF does not define which flow line is joining which. The following two GR examples can produce the same SDL-CIF. (Example PR without flow lines.)



```
/* CIF Label (150,100),(100,100) */
connection
  one :
  /* CIF Output (100,300) */
  output alternativeOne;
  /* CIF Label Invisible */
  grs0 :
  /* CIF NextState (250,500) */
  nextstate continue;
/* CIF End Label */
endconnection one;
/* CIF Label (450,100),(100,100) */
connection
  two :
  /* CIF Output (400,300) */
  output alternativeTwo;
  /* CIF Join Invisible */
  join grs0;
/* CIF End Label */
endconnection two;
```

7.2.4 Lines and enclosing rectangles

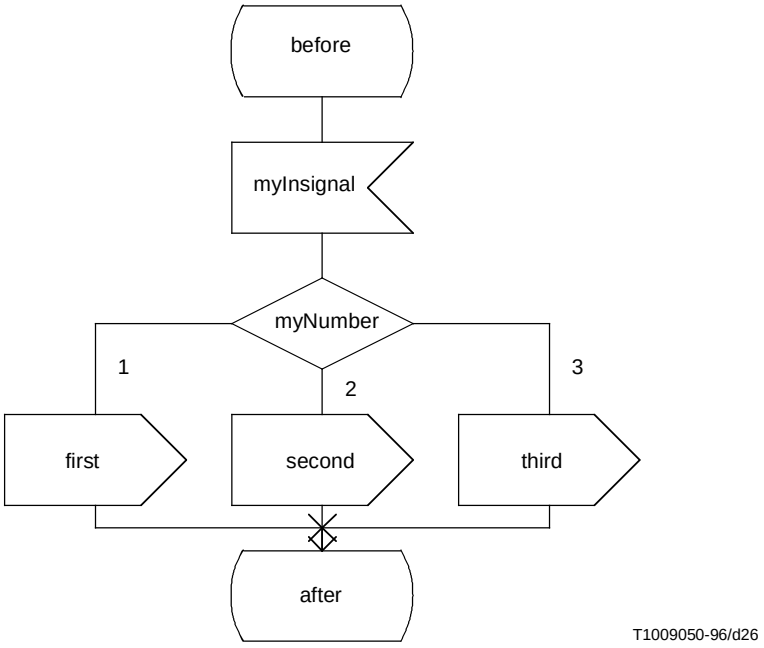


T1009040-96/d25

Note that the text extension line is described as if its endpoints were on the border of the surrounding rectangles of the attached symbols even if this is not completely true in GR. (The endpoint on the input symbol is inside the surrounding rectangle.) This rule applies to all lines/signal routes/channels connected to symbols that are not rectangular shaped. (Stop symbol, process symbol...)

```
/* CIF State (100,100) */
state myStart;
  /* Input (100,300) */
  input mySignal
/* CIF TextExtension (400,300) */
/* CIF Line (400,350),(300,350) */
(myParameter)
/* CIF End TextExtension */
;
    /* CIF NextState (100,500) */
    nextstate myNextState;
```

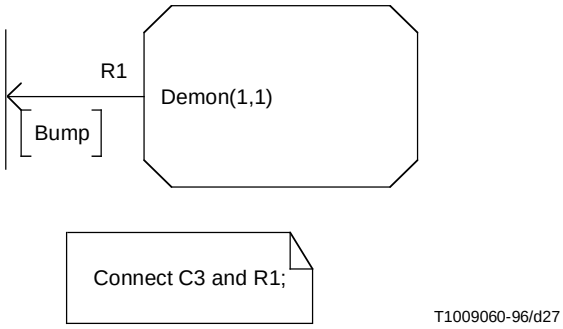
7.2.5 Answer flow lines after decision




```
/* CIF State (400,100) */
state before;
  /* CIF Input (400,300) */
  input myInsignal;
    /* CIF Decision (400,500) */
    decision myNumber;
    /* CIF Answer Left InvisibleBrackets */
    (1) :
      /* CIF Output (100,700) */
      output First;
    /* CIF Answer InvisibleBrackets */
    (2) :
      /* CIF Output (400,700) */
      output second;
    /* CIF Answer Right InvisibleBrackets */
    (3) :
      /* CIF Output (700,700) */
      output third;
    /* CIF End Decision */
    enddecision;
    /* CIF NextState (400,900) */
    nextstate after;
/* CIF End Estate */
endstate;
```

7.2.6 Connect information in text symbols vs. near the frame symbol

The connect information can be specified in two ways in GR. Both ways produce the same PR, we have different CIF comments to distinguish between the two:



T1009060-96/d27

```
/* CIF Text (200,800) */
Connect C3 and R1;
/* CIF TextEnd */
```

T1009060-96/d28

```
/* CIF Connect */
/* CIF TextPosition (25,700) */
Connect C3 and R1;
```

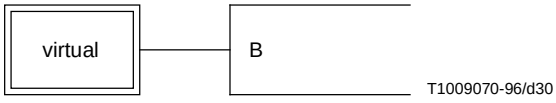
7.2.7 Text extension

This is allowed:

T1009070-96/d29

```
/* CIF Process (800,550) */
PROCESS Demon
/* CIF TextExtension (1100,550) */
(1,1)
/* CIF TextExtensionEnd */
REFERENCED;
```

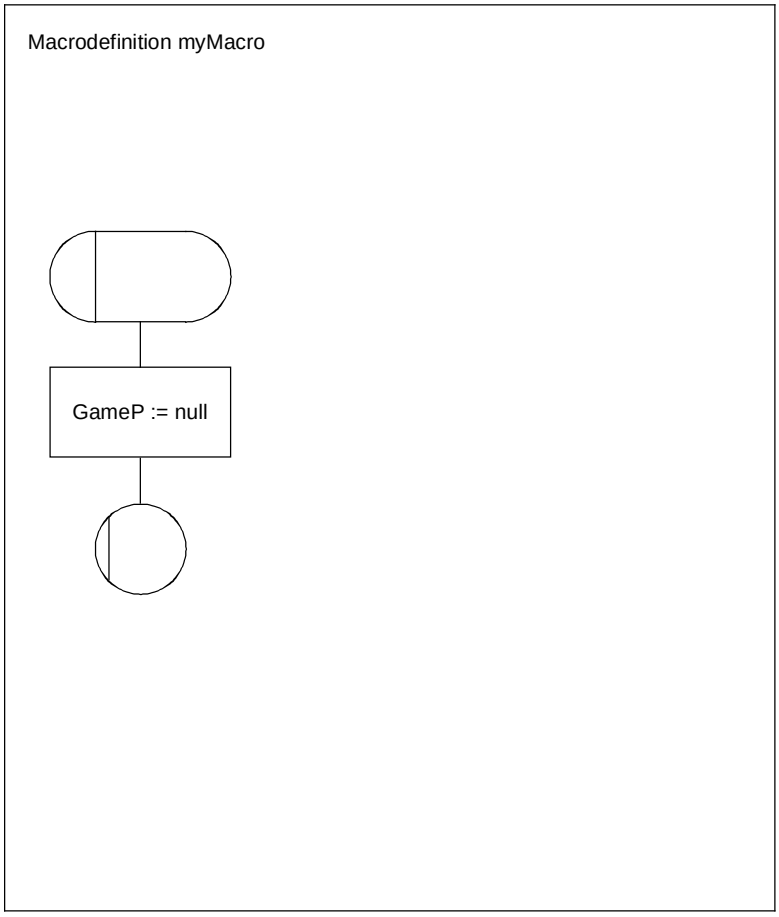
This is also allowed:



T1009070-96/d30

```
/* CIF BlockType (800,550) */
virtual BLOCK TYPE
/* CIF TextExtension (1100,550) */
B
/* CIF TextExtensionEnd */
REFERENCED;
```

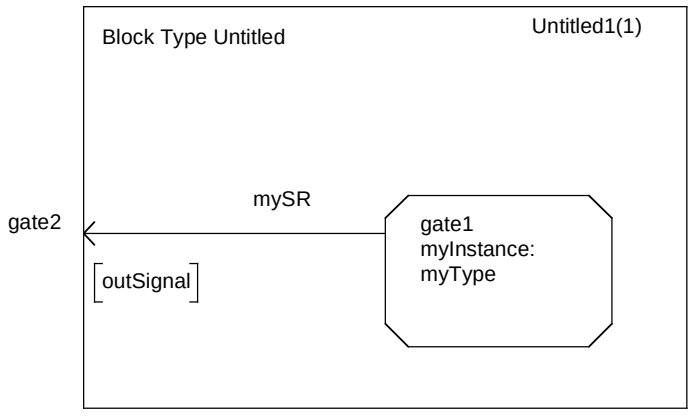
7.2.8 Macro diagram



T1009080-96/d31

```
/* CIF MacroDiagram */
/* CIF Page 1 (1000,1000) */
Macrodefinition myMacro;
/* CIF MacroInlet (300,100) */
/* CIF Task (300,300) */
TASK GameP := null;
/* CIF MacroOutlet (300,500) */
/* CIF End MacroDiagram */
ENDMACRO myMacro;
```

7.2.9 Text positions for gate references

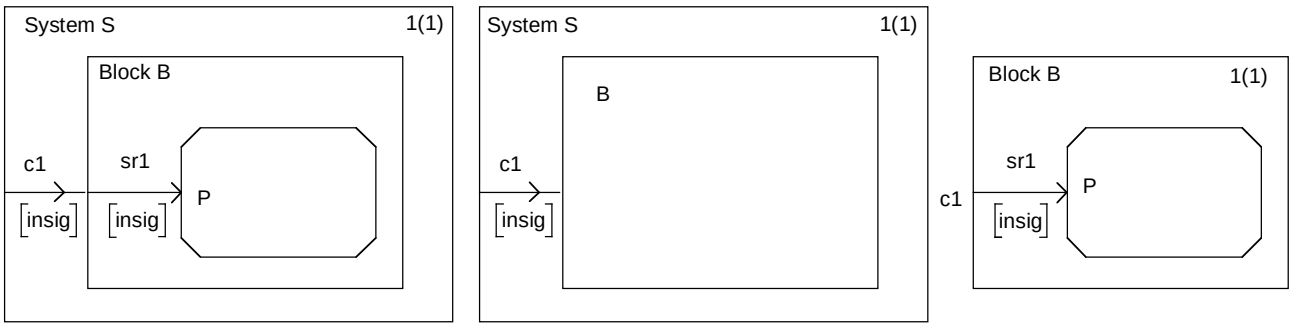


T1009090-96/d32

Note that the text position for gate2 is defined in connection to the page/frame definition while the text gate2 is defined in the PR for the signal route. In a similar way, the text position for gate1 is defined in connection with the process symbol while the text gate1 is defined in the PR for the signal route. To find out which gate reference text position goes with which text, the tool has to match the gate reference connection positions with signal route end points.

```
/* CIF BlockTypeDiagram */
/* CIF Page Untitled1 (1900,1100) */
/* CIF Frame (100,100),(800,700) */
/* CIF GateReference (100,550) */
/* CIF TextPosition (10,540) */
Block Type Untitled;
/* CIF Process (300,500),(200,100) */
/* CIF GateReference (300,550) */
/* CIF TextPosition (310,540) */
PROCESS myInstance:
myType;
/* CIF SignalRoute (300,550),(100,550) */
/* CIF TextPosition (200,500) */
/* CIF TextPosition (130,570) SignalRoute1 */
SIGNALROUTE mySR
FROM myInstance VIA gate1 TO ENV VIA gate2 WITH outSignal;
/* CIF End BlockTypeDiagram */
ENDBLOCK TYPE;
```

7.2.10 Nested diagrams



T1009100-96/d33

This nested diagram is converted into two diagrams with the following steps:

1) Replace the block diagram in system S with a block reference symbol.
2) Create a separate block diagram.
3) Insert information about what channel the signal route is connected to.
4) Name the page with an appropriate name.

a) *Nested form*

```
/* CIF SystemDiagram */
/* CIF Page 1 (600,500) */
System S;
/* CIF Channel (0,250),(100,250) */
CHANNEL c1
FROM ENV TO B WITH insig;
/* CIF BlockDiagram */
/* CIF NestedFrame (100,100),(400,300) */
Block B;
  /* CIF SignalRoute (100,250),(200,250) */
  SIGNALROUTE sr1
  FROM ENV TO P WITH insig;
```

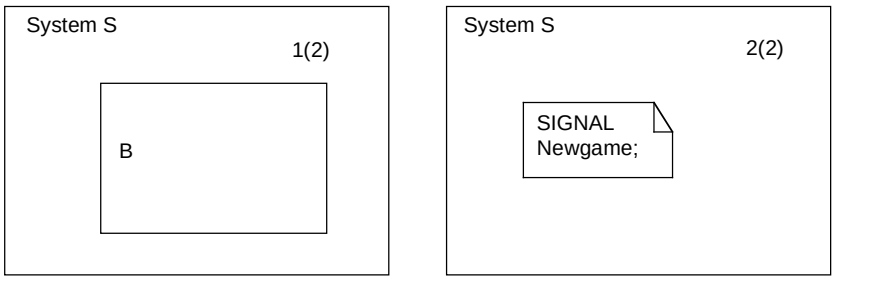
```
  /* CIF Connect */
  CONNECT c1 AND sr1;
  /* CIF Process (200,200),(200,100) */
  PROCESS P REFERENCED;
/* CIF End BlockDiagram */
ENDBLOCK B;
/* CIF End SystemDiagram */
ENDSYSTEM S;
```

b) *After conversion/without nesting*

```
/* CIF SystemDiagram */
/* CIF Page 1 (600,500) */
System S;
/* CIF Channel (0,250),(100,250) */
CHANNEL c1
FROM ENV TO B WITH insig;
/* CIF Block (100,100),(400,300) */
BLOCK B REFERENCED;
/* CIF End SystemDiagram */
ENDSYSTEM S;
```

```
/* CIF BlockDiagram */
/* CIF Page 1 (600,500) */
/* CIF Frame (100,100),(400,300) */
Block B;
/* CIF SignalRoute (100,250),(200,250) */
SIGNALROUTE sr1
FROM ENV TO P WITH insig;
/* CIF Connect */
CONNECT c1 AND sr1;
/* CIF Process (200,200),(200,100) */
PROCESS P REFERENCED;
/* CIF End BlockDiagram */
ENDBLOCK B;
```

7.2.11 Many pages



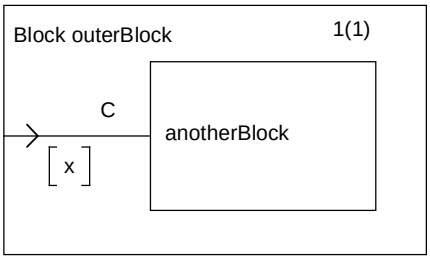
T1009110-96/d34

```
/* CIF SystemDiagram */
/* CIF Page 2 (600,300) */
/* CIF Page 1 (600,300) */
System S;
/* CIF DefaultSize (200,100) */
/* CIF Block (200,100) */
BLOCK B REFERENCED;
/* CIF CurrentPage 2 */
/* CIF Text (200,100) */
SIGNAL
NewGame;
/* CIF End Text */
/* CIF End SystemDiagram */
ENDSYSTEM S;
```

7.2.12 Block in block

People often place block reference symbols in block diagrams in SDL-GR. This is a shorthand that is not allowed in SDL-PR. When generating PR from the GR, the block in block shorthand is expanded to block in substructure in block. Here are some examples showing how CIF handles this and similar situations. CIF comments for text positions are left out to make the text easier to read:

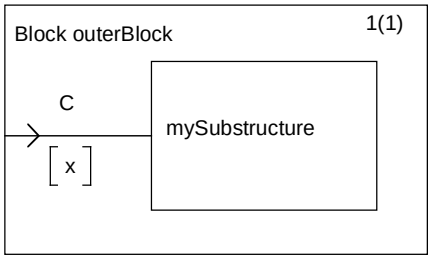
Block reference symbol in block diagram



T1009120-96/d35

```
/* CIF BlockDiagram */
/* CIF Page 1 (400,300) */
BLOCK outerBlock;
  /* CIF SubstructureDiagram Invisible */
  SUBSTRUCTURE outerBlock;
    /* CIF Channel (0,150),(100,150) */
    CHANNEL C FROM env TO anotherBlock WITH x;
    ENDCHANNEL C;
    /* CIF Block (100,100),(200,100) */
    BLOCK anotherBlock REFERENCED;
  /* CIF End SubstructureDiagram */
  ENDSUBSTRUCTURE outerBlock;
/* CIF End BlockDiagram */
ENDBLOCK;
```

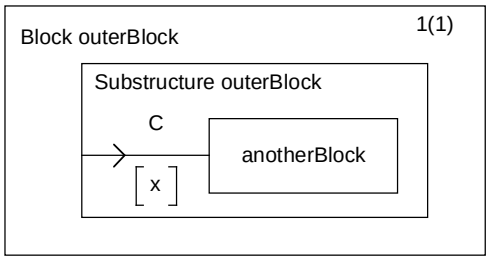
Substructure reference symbol in block diagram



T1009130-96/d36

```
/* CIF BlockDiagram */
/* CIF Page 1 (400,300) */
BLOCK outerBlock;
  /* CIF BlockSubstructure (100,100),(200,100) */
  SUBSTRUCTURE mySubstructure REFERENCED;
  /* CIF Channel (0,150),(100,150) */
  CHANNEL C FROM env TO anotherBlock WITH x;
  ENDCHANNEL C;
/* CIF End BlockDiagram */
ENDBLOCK;
```

Nested substructure diagram in block diagram



T1009140-96/d37

Note that the SDL-PR in this example is exactly the same as in the first example in this subclause while the SDL-GR is different.

```
/* CIF BlockDiagram */
/* CIF Page 1 (600,500) */
BLOCK outerBlock;
  /* CIF SubstructureDiagram */
  /* CIF NestedFrame (100,100),(400,300) */
  SUBSTRUCTURE outerBlock;
    /* CIF Channel (100,250),(200,250) */
    CHANNEL C FROM env TO anotherBlock WITH x;
    ENDCHANNEL C;
    /* CIF Block (200,200),(200,100) */
    BLOCK anotherBlock REFERENCED;
  /* CIF End SubstructureDiagram */
  ENDSUBSTRUCTURE outerBlock;
/* CIF End BlockDiagram */
ENDBLOCK;
```

7.3 Situations CIF cannot handle

T1009150-96/d38

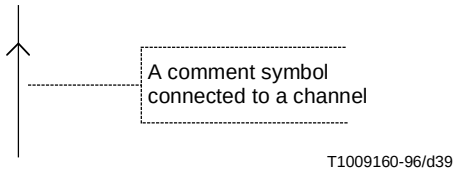
CIF cannot handle this properly. The CIF code would be:

```
/* CIF Process (800,550) Dashed myProcess */
```

The comment in the GR is lost. The consequence is that CIF cannot be used as a storage format for all legal GR diagrams. To manage this example properly, the complete text within the symbol has to be stored inside the CIF comment. This would mean that escape characters had to be introduced to manage start of comment, end of comment, start of text and end of text tokens/characters within the text. However, this has been left out of CIF to avoid too much complexity.

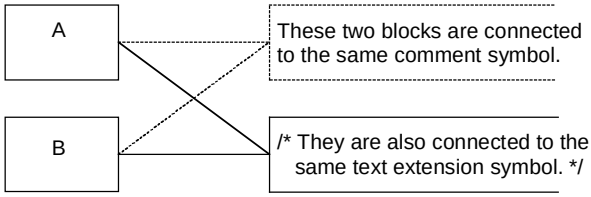
This is one example from the class of problems that arise because SDL-PR is not matching SDL-GR exactly. Problems with expressing textual comments in PR can also be found in connection with, for instance, connect, gates and macros.

Here is another example of problem with comments. This time it is a comment symbol. How would you express this in PR?



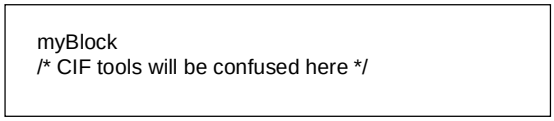
T1009160-96/d39

Here is one more SDL-GR example, that may or may not be legal SDL-GR, that neither CIF nor PR at the moment can handle well. (The most reasonable thing to do when converting this to PR is to duplicate the text extension/comment symbol. What should then happen when we are converting back to GR?)



T1009170-96/d40

There is one situation that CIF cannot handle because of the construction of CIF itself. It is if the SDL-GR contains something that resembles CIF:



T1009180-96/d41

8 CIF conformance criteria

8.1 About tools reading a CIF file

A tool reading a CIF file should try to be forgiving instead of following the rules strictly.

Example 1: The CIF file says that the current page is X, but there is no page X in the current diagram. The tool should issue a warning and ignore the erroneous current page CIF comment.

Example 2: The CIF file describes a flow line that does not end on the border of a symbol in both ends. The tool should issue a warning and ignore the erroneous flow line CIF comment.

8.2 Automatic vs. forced layout

When a CIF file is exported from a tool, it contains positioning information that relates to the layout of the SDL diagrams. When another tool imports this information, it will make use of the layout information in the CIF file. If the tool does not use the layout information and uses an automatic layout mechanism, then it conforms to the level 1 CIF for import only. Tools that support CIF should clearly indicate whether they use forced layout when importing CIF and how they support the preservation of graphical information and to which CIF levels they conform for export and for import.

8.3 Retainment and use of tool-specific information

The CIF has been intentionally defined to provide tool-specific information that may or may not be used by another tool. ITU-T recommends that, in case tool vendors use tool-specific information, they provide this information publicly to avoid a clash with other tools. An example has been provided in this Recommendation and future CIF versions may include additional tool-specific information to enable other tool vendors to support the import and possible preservation of this information. No license fees shall be applicable to any tool-specific information that uses the facilities for tool-specific information provided by this Recommendation.

CIF keyword index

Note that the index does not include tool-specific CIF comments.

K
Keep <tool-specific CIF comment: C0>

L
Label <label symbol: A64>, <descriptor end: A70>
Left <text extension: A25>, <comment: A26>, <answer flow line: A29>, <input symbol: A65>, <priority input symbol: A66>, <output symbol: A67>
Line <line: B20>, <dashed line: B21>

M
MacroCall <macro call symbol: A61>
MacroDiagram <macro diagram start: A16>, <diagram end: A17>
MacroInlet <macro inlet symbol: A62>
MacroOutlet <macro outlet symbol: A63>

N
NestedFrame <nested frame: B13>
NextState <nextstate symbol: A46>

O
Operator <operator symbol: A42>
OperatorDiagram <operator diagram start: A15>, <diagram end: A17>
OutletText <outlet text: B11>
Output <output symbol: A67>

P
PackageDiagram <package diagram start: A4>, <diagram end: A17>
PackageReference <package reference: B18>
Page <page type: B12>
PageName <page text position: B23>
PriorityInput <priority input symbol: A66>
Procedure <procedure symbol: A41>
ProcedureCall <procedure call symbol: A53>
ProcedureDiagram <procedure diagram start: A14>, <diagram end: A17>
ProcedureStart <procedure start symbol: A54>
Process <dashed process symbol: A33>, <process symbol rectangle: B16>
ProcessDiagram <process diagram start: A10>, <diagram end: A17>
ProcessType <process type symbol: A38>
ProcessTypeDiagram <process type diagram start: A11>, <diagram end: A17>

R
Reset <reset symbol: A50>
Return <return symbol: A55>
Right <text extension: A25>, <comment: A26>, <answer flow line: A29>, <input symbol: A65>, <priority input symbol: A66>, <output symbol: A67>

S
Save <save symbol: A47>
Select <select symbol: A69>, <descriptor end: A70>
Service <dashed service symbol: A35>, <service symbol rectangle: B17>
ServiceDiagram <service diagram start: A12>, <diagram end: A17>
ServicePage <page type: B12>
ServiceType <service type symbol: A39>
ServiceTypeDiagram <service type diagram start: A13>, <diagram end: A17>
Set <set symbol: A49>
SignalList1 <first signallist text position: B6>
SignalList2 <second signallist text position: B7>
SignalRoute <signal route: A22>

Specific <tool-specific CIF comment: C0>
Start <start symbol: A43>
State <state symbol: A45>, <descriptor end: A70>
Stop <stop symbol: A44>
SubstructureDiagram <substructure diagram start: A9>, <diagram end: A17>
SystemDiagram <system diagram start: A5>, <diagram end: A17>
SystemType <system type symbol: A36>
SystemTypeDiagram <system type diagram start: A6>, <diagram end: A17>

T
Task <task symbol: A48>
Text <text symbol: A68>
TextExtension <text extension: A25>
TextPosition <first signallist text position: B6>, <second signallist text position: B7>, <page text position: B23>, <text position: B25>
TransitionOption <transition option symbol: A59>, <descriptor end: A70>

Appendix I

Tool-specific CIF comments

I.1 Maintenance of CIF

As it is totally impossible to foresee in SDL-CIF all potential requests coming from tool makers regarding tool-specific directives, the syntax has been opened up to continuously integrate new tool-specific directives.

First of all, names of new tools must be approved by the ITU-T Z.106 Working Group, to ensure that there is no name conflict.

In the perspective of enriching SDL-CIF, tool makers who have created new tool-specific directives should propose them to the Z.106 Working Group. In this way, these directives have a chance to be introduced in this Recommendation, as new non-specific recommended directives, either mandatory or optional.

Successful and unsuccessful implementations of SDL-CIF should be reported to the Z.106 Working Group, in order to adjust to mandatory/optional classification of directives.

An up-to-date list of tool-specific CIF comments in use by various tools is available on the ITU web site. In addition, this Appendix provides an initial list of tool-specific CIF comments that were known when this Appendix was produced.

I.2 Current tool-specific CIF comments

The tool-specific CIF comments defined below are not a part of the CIF standard. They constitute the currently known tool-specific CIF comments.

How a tool should handle tool-specific CIF comments is defined in the subclause about tool-specific CIF comments in this Recommendation.

In this Appendix we will use an imaginary SDL tool which has <tool name> equal to mySDLTool; a tool developer should substitute mySDLTool for an appropriate name that uniquely identifies the tool in question. The tool-specific keywords defined in this Appendix cannot be found in the keyword index of this Recommendation.

I.2.1 Placement of tool-specific CIF comments

This topic is discussed in the additional information associated with <tool-specific CIF comments: C0>.

Example

In the example below, many tool-specific CIF comments are associated with the CIF rule <system diagram: A5>. Detailed information for each tool-specific CIF comment mentioned in this example can be found later in this Appendix.

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Page 2 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool Version 1.0 */
/* CIF Specific mySDLTool Page 1 Scale 200 AutoNumbered
FixedHeadingSize (200,100) */
/* CIF Specific mySDLTool Page 2 Scale 200 AutoNumbered */
/* CIF Specific mySDLTool OriginalFileName 'mysystem.ssy'*/
SYSTEM mySystem;
/* CIF CurrentPage 1 */
/* CIF Text (800,550),(200,100) */
/* CIF Specific mySDLTool FixedSize (200,100) */
dcl
  MyNo Integer;
timer
  DoorTimer;
/* CIF End Text */
...
```

C1 tool version number:

/* CIF Specific mySDLTool Version x.y */
<diagram description: A2>

Association:

This tool-specific CIF comment is associated with an A rule mentioned in the rule for <diagram start: A3>.

Example (where this tool-specific CIF rule is associated with <system diagram start: A5>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool Version 1.2 */
SYSTEM mySystem;
```

C2 original file:

/* CIF Specific mySDLTool OriginalFileName <file name char string literal> ***/**
<diagram start: A3>

Association:

This tool-specific CIF comment is associated with an A rule mentioned in the rule for <diagram start: A3>.

Additional information:

This comment is used to remember the file name of the binary file that was converted to CIF. When the CIF file is converted to a binary file again, the same name can be reused. Only the file name (a.ssy) will be given in <file name char string literal>, not the complete path (/home/lat/a.ssy).

Example (where this tool-specific CIF rule is associated with <system diagram start: A5>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool OriginalFileName 'mysystem.ssy' */
SYSTEM mySystem;
```

C3 page details specification:

/* CIF Specific mySDLTool Page <page number> [**ShowMeFirst**] [**Scale** <integer>] [**Grid <**point: B26>] [**AutoNumbered**] [**FixedHeadingSize** [<fixed size point: B26>]] ***/**

Association:

This tool-specific CIF comment is associated with an A rule mentioned in the rule for <diagram start: A3>. There may be zero or one of this tool-specific CIF comment for each <page declaration: B3> in the diagram.

Additional information:

This tool-specific CIF comment is used to give additional information for an already defined page:

- If this page should pop up as the default page when the diagram is loaded into an editor (The keyword ShowMeFirst is given.) or not (The keyword ShowMeFirst is not given.).
- Specify the scale used in the SDL editor when presenting the diagram. Scale is expressed as per cent of normal size. If scale is not given, 100% is used.
- The grid size for the page. If grid is not given, grid will be (50,50) which is the same as no grid.
- If autonumbering should be used (The keyword AutoNumbered is given.) or not (The keyword AutoNumbered is not given.). AutoNumbered means that the tool will keep track of page numbering in the following way: Initially, the first declared autonumbered page will get the number "1", the second declared autonumbered page will get the number "2"... Later when the user adds an autonumbered page before page "2", the new page will get the number "2" and all old autonumbered pages after page "1" will get their name increased by one. A unique <page name charstring literal> must be given in <page declaration: B3> for the page even if the page is autonumbered to be able to refer to the page from other CIF comments.

In addition, if FixedHeadingSize is given, this tool-specific CIF comment says that the additional heading symbol should be given a fixed size and not be adjusted to the size of the text. This means that the complete text will not be visible. The user has to double click on the additional heading symbol to expand the symbol to the larger size. The SDL tool uses this comment for additional heading symbols that the user has "collapsed" with a double click.

Example (where this tool-specific CIF rule is associated with <system diagram start: A5>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Page 2 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool Page 1 Scale 200 AutoNumbered */
/* CIF Specific mySDLTool Page 2 Scale 200 AutoNumbered */
SYSTEM mySystem;
/* CIF Text (800,550) */
SIGNAL Newgame;
/* CIF End Text */
```

C4 fixed size:

/* CIF Specific mySDLTool FixedSize [<fixed size point: B26>] ***/**

Association:

This tool-specific CIF comment is associated with <text symbol: A68>.

Additional information:

This tool-specific CIF comment says that although the symbol we have just defined in a CIF comment was given a size (a size that makes the complete text visible), we use this fixed smaller size to render the symbol. This means that the complete text will not be visible. The user has to double click on the symbol to expand the symbol to the larger size. The SDL tool uses this comment for text symbols that the user has "collapsed" with a double click.

Example:

```
/* CIF Text (800,550),(200,200) */
/* CIF Specific mySDLTool FixedSize (200,100) */
dcl
  MyNo Integer;
timer
  DoorTimer;
/* CIF End Text */
```

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communication
Series Z	**Programming languages**