ITU - Telecommunication Standardization Sector **Temporary Document SAxx-E**

STUDY GROUP 10 **Original: English**

Sophia Antipolis, 98.10.20 – 98.10.23

Question: 9 (MSC)

SOURCE: RAPPORTEUR

TITLE: EMPTY BEHAVIOUR IN SHARED INLINE EXPRESSIONS

Contact: Jan Docekal, Telelogic AB, jan.docekal@telelogic.com

Abstract

This is a correction to shared inline expressions so that they will allow empty behavior.

Problem description

alt expr is taken for example

```
<shared alt expr> ::=
   alt begin [ <inline expr identification> ] <shared> <end>
     [ <inline gate interface> ]
     <instance event list>
   {
     alt <end>
     [ <inline gate interface> ]
     <instance event list>
   } *
   alt end <end> .

<instance event list> ::= <instance event> + .
```

This syntax don't allow to specify *empty* behavior for some instance. In some cases empty behavior is needed:

```
a : alt begin shared b;
      action 'x:=1';
    alt;
    alt end;
b : alt begin shared a;
    alt;
      action 'x:=2';
    alt end;
```

This MSC in event oriented syntax “looks” like:

```
a, b : alt begin;
```

  a : **action** 'x:=1';
alt;
  b : **action** 'x:=2';
alt end;

and neither alternative *can* be empty. However if in the instance oriented syntax empty alternatives *can* appear.

Solution

The following changes are necessary, since the non shared version of inline expressions allows empty behavior for all inline expressions.

Change (shared inline expr): Correction of Shared Inline Expression

<shared loop expr> ::=
loop [<loop boundary>] **begin** [<inline expr identification>]<shared> <end>
[<inline gate interface>] <instance event list>
loop end <end>

<shared opt expr> ::=
opt begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] <instance event list>
opt end <end>

<shared exc expr> ::=
exc begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] <instance event list>
exc end <end>

<shared alt expr> ::=
alt begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] <instance event list>
{ **alt** <end> [<inline gate interface>] <instance event list> }*
alt end <end>

<shared par expr> ::=
par begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] <instance event list>
{ **par** <end> [<inline gate interface>] <instance event list> }*
par end <end>

<shared loop expr> ::=
loop [<loop boundary>] **begin** [<inline expr identification>]<shared> <end>
[<inline gate interface>] [<instance event list>]
loop end <end>

<shared opt expr> ::=
opt begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] [<instance event list>]
opt end <end>

<shared exc expr> ::=
exc begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] [<instance event list>]

exc end <end>

<shared alt expr> ::=

alt begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] [<instance event list>]
{ **alt** <end> [<inline gate interface>] [<instance event list>] }*
alt end <end>

<shared par expr> ::=

par begin [<inline expr identification>] <shared> <end>
[<inline gate interface>] [<instance event list>]
{ **par** <end> [<inline gate interface>] [<instance event list>] }*
par end <end>