**Output Channel Error Handling Algorithm CEH.GD.Auto**

(Formalized Specification)

**Pi::CEH.GD.Auto**

**{SYNOPSIS}**

The algorithm is used to automatically keep the output connection open. Supports "graceful degradation" in combination with the algorithm RUP.GD for communication ring integrity checkup.

The algorithm monitors the state of the output channel to the immediate neighbor and is responsible for its opening. Starts just after process creation as well as at the process output disconnection with its immediate neighbor.

**{ASSUMPTIONS}**

The CEH.GD.Auto algorithm is valid if the following conditions are met:

* The distributed system is synchronous.
* The type of process failures is “fail-stop”; however, is allowed the process auto reparation before it is finally marking as faulty.

**ALGORITHM 1**: Declarative Part of *Pi::CEH.GD.Auto*

**{SYSTEM CONSTANTS}**

Int MAX\_CEH\_PERIOD // period between reconnection attempts

Int MAX\_CEH\_ERR // maximum reconnection attempts

Int MIN\_K // min number of faultless processes (max degradation), 1 ≤ MIN\_K

PId i // process Pi identifier

PId j // default neighbor process Pj

Bool AutoList // <true> if ListPIds is to be filled during RUP.GD.Auto

// <false> if ListPIds is known in advance (RUP used)

**{MESSAGES}**

**{SET OF STATES}**

<State> := {INIT, CLOSED, OPENED, FAULTY}

**{INTERNAL STATE SPACE}**

State state // current process CEH state

Int ErrorCounter // error counter

Timer TimerCEH // timer

List ListPIds // list of process identifiers

PId PIdNext // current neighbor process identifier

**ALGORITHM 2**: Event Handlers of *Pi::CEH.GD.Auto*

**OnInit:**

state := INIT

ErrorCounter := 0

TimerCEH.Interval := MAX\_CEH\_PERIOD

ListPIds.Set()

**If** AutoList = true

ListPIds.Clear()

PIdNext := j

**Else**

ListPIds.Set()

PIdNext := ListPIds.PopFront()

**End If**

**OnShow:**

ErrorCounter := 0

ChannelOut.Open()

**OnOutputConnect:**

state := OPENED

ErrorCounter := 0

*{Start Ring Check Up Algorithm}*

**OnOutputDisconnect:**

state := CLOSED

ErrorCounter := 0

TimerCEH.Start()

**OnOutputError:**

state := CLOSED

ErrorCounter := ErrorCounter + 1

**If** ErrorCounter < MAX\_CEH\_ERR

TimerCEH.Start()

**Else**

**If** ((AutoList = false) (RUP::state = UP)) ∩ (ListPids.Size() ≥ MIN\_K)

PIdNext := ListPIds.PopFront()

ErrorCounter := 0

TimerCEH.Start()

**Else**

*{UNRECOVERABLE FAILURE}*

**End If**

**End If**

**OnTimer**:

TimerCEH.Stop()

ChannelOut.Open()