АЛГОРИТЪМ ЗА ОБРАБОТКА НА ГРЕШКИТЕ ПО ИЗХОДНИЯ КАНАЛ

Output Channel Error Handling Algorithm *CEH.GD*  
(Formalized Specification)

**Pi::CEH.GD**

**{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 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.**

**{SYSTEM CONSTANTS}**

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

Int MAX\_CEH\_ERR // maximum connection attempts

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

PId i // process Pi identifier

PId j // default neighbor process Pj

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

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

**{SET OF STATES}**

<State> := {CEH\_INIT, CEH\_CLOSED, CEH\_OPENED, CEH\_FAULTY}

**{INTERNAL STATE SPACE}**

State state // current process state

Int ErrorCounter // error counter

Timer TimerCEH // timer

List ListPIds // list of process identifiers

PId PIdNext // current neighbor process identifier

**{EVENTS}**

**OnInit:**

state := CEH\_INIT

ErrorCounter := 0

TimerCEH.Interval := MAX\_CEH\_PERIOD

**If** AutoList = true

ListPIds.Clear()

**Else**

ListPIds.Set()

**EndIf**

PIdNext := j

**OnShow:**

ErrorCounter := 0

ChannelOut.Open()

**OnOutputConnect:**

state := CEH\_OPENED

ErrorCounter := 0

{*Start Ring Check Up Algorithm*}

**OnOutputDisconnect:**

state := CEH\_CLOSED

ErrorCounter := 0

TimerCEH.Start()

**OnOutputError:**

state := CEH\_CLOSED

ErrorCounter := ErrorCounter + 1

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

TimerCEH.Start()

**Else**

**If** ListPids.Size() MIN\_K

PIdNext := ListPIds.PopFront()

ErrorCounter := 0

TimerCEH.Start()

**Else**

{*UNRECOVERABLE FAILURE*}

**EndIf**

**EndIf**

**OnTimer:**

TimerCEH.Stop()

ChannelOut.Open()