АЛГОРИТЪМ ЗА ОБРАБОТКА НА ГРЕШКИТЕ ПО ИЗХОДНИЯ КАНАЛ

Output Channel Error Handling Algorithm *CEH*  
(Generalized Formal Specification[[1]](#footnote-1))

**Pi::CEH**

**{SYNOPSIS}**

**The algorithm is used to automatically keep the output connection open.**

**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

PId i // process Pi identifier

**{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

**{EVENTS}**

**OnInit:**

state := CEH\_INIT

ErrorCounter := 0

TimerCEH.Interval := MAX\_CEH\_PERIOD

**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**

{*UNRECOVERABLE FAILURE*}

**EndIf**

**OnTimer:**

TimerCEH.Stop()

ChannelOut.Open()

1. Implementation <https://github.com/milphaser/XME.Ring> [↑](#footnote-ref-1)