**Ring-Based Distributed Mutual Exclusion (ME)**

(Formalized Specification)

**Pi::ME**

**{SYNOPSIS}**

Basic variant of the algorithm for distributed mutual exclusion over logical communication ring.

**{ASSUMPTIONS}**

No failures of any kind (neither processes nor channels) are allowed, by condition.

Only under this condition the two mandatory requirements for this class of distributed algorithms - ME1 (safety) and ME2 (feasibility, liveness) are guaranteed. The desirable, but not mandatory, condition ME3 (fairness) is not met for this algorithm by definition.

**ALGORITHM 1**: Declarative Part of *Pi::ME*

**{SYSTEM CONSTANTS}**

String MRK\_ME // type of message “ME token”

**{MESSAGES}**

<mrk\_me>

**{SET OF STATES}**

<State> := {INIT, RELEASED, WANTED, HELD}

**{INTERNAL STATE SPACE}**

State state // current process ME state

E::pidCoordinator // “coordinator” process identifier

**ALGORITHM 2**: Event Handlers of *Pi::ME*

**OnInit:**

state := INIT

E::pidCoordinator := NULL

**OnRingUp:**

E::OnStart()

**OnAfterElection:**

**If** i = E::pidCoordinator

StartMarker()

**End If**

state := RELEASED

**OnEnter:**

**If** state = RELEASED

state := WANTED

**End If**

**OnReceiptOf <mrk\_me>:**

**If** state = WANTED

state:= HELD

**Else If** state = RELEASED

Send <mrk\_me>

**End If**

**OnRelease:**

**If** state = HELD

state := RELEASED

Send <mrk\_me>

**End If**