

=====
An Event-B Specification of CSW_modes @ Date: 31 Oct 2008 @ Time: 01:30:15 PM
=====

MACHINE CSW_modes

REFINES Core_SW

SEES CSW_modes_context

VARIABLES

TCbuffer
TMbuffer
TCpool
TMpool
TCstatus
csw_hk_on
csw_sci_on
csw_mode

INVARIANTS

inv1 : *csw_mode* \in *CSW_MODES*

EVENTS

INITIALISATION

BEGIN

act1 : *TCbuffer* := \emptyset
act2 : *TMbuffer* := \emptyset
act3 : *TCpool* := \emptyset
act4 : *TMpool* := \emptyset
act5 : *TCstatus* := \emptyset
act6 : *csw_hk_on* := *FALSE*
act7 : *csw_sci_on* := *FALSE*
act8 : *csw_mode* := *csw_standby*

END

EVENT env_Receive_TC

REFINES env_Receive_TC

ANY

tc
tc_pointer

WHERE

grd1 : *tc* \in *TC_SET*
grd2 : *tc_pointer* \in *TC_BUFFER_ADDRESSES*
grd3 : *tc_pointer* \notin *dom(TCbuffer)*

THEN

act1 : *TCbuffer*(*tc_pointer*) := *tc*

END

EVENT Poll_TC

REFINES Poll_TC

ANY

tc_pointer

```

    tc_handler
WHERE
    grd1 : tc_pointer ∈ dom(TCbuffer)
    grd2 : tc_handler ∈ TCTM_POOL_ADDRESSES
    grd3 : tc_handler ∉ dom(TCpool)
    grd4 : tc_handler ∉ dom(TMpool)
THEN
    act1 : TCpool(tc_handler) := TCbuffer(tc_pointer)
    act3 : TCstatus(tc_handler) := Unchecked
    act2 : TCbuffer := {tc_pointer} ⧹ TCbuffer
END

EVENT Report_TC_check_ok
REFINES Report_TC_check_ok
ANY
    tc_handler
    tm_handler
    tm
WHERE
    grd1 : tc_handler ∈ dom(TCpool)
    grd2 : tm_handler ∈ TCTM_POOL_ADDRESSES
    grd3 : tm ∈ TM_SET
    grd4 : tm_handler ∉ dom(TCpool)
    grd5 : tm_handler ∉ dom(TMpool)
    grd6 : TCstatus(tc_handler) = Unchecked
    grd7 : Valid_TC(TCpool(tc_handler)) = TRUE
    grd8 : Type_of_TM(tm) = Check_ok_TM
THEN
    act1 : TCstatus(tc_handler) := Waiting_for_Execution
    act2 : TMpool(tm_handler) := tm
END

EVENT Report_TC_check_fail
REFINES Report_TC_check_fail
ANY
    tc_handler
    tm_handler
    tm
WHERE
    grd1 : tc_handler ∈ dom(TCpool)
    grd2 : tm_handler ∈ TCTM_POOL_ADDRESSES
    grd3 : tm ∈ TM_SET
    grd4 : tm_handler ∉ dom(TCpool)
    grd5 : tm_handler ∉ dom(TMpool)
    grd6 : TCstatus(tc_handler) = Unchecked
    grd7 : Valid_TC(TCpool(tc_handler)) = FALSE
    grd8 : Type_of_TM(tm) = Check_fail_TM
THEN
    act1 : TCpool := {tc_handler} ⧹ TCpool
    act3 : TCstatus := {tc_handler} ⧹ TCstatus
    act2 : TMpool(tm_handler) := tm
END

EVENT TC_Execution_Mode_change
REFINES TC_Execution_Mode_change
ANY
    tc_handler

```

new_mode
WHERE
 $grd1 : tc_handler \in dom(TCpool)$
 $grd2 : TCstatus(tc_handler) = Waiting_for_Execution$
 $grd3 : Type_of_TC(TCpool(tc_handler)) \in MODE_CHANGING_TC$
 $grd4 : new_mode \in CSW_MODES$
 $grd5 : (csw_mode \mapsto new_mode) \in CSW_TC_TRANSITIONS$
THEN
 $act1 : TCstatus(tc_handler) := Successfully_Executed$
 $act3 : csw_mode := new_mode$
END

EVENT TC_ASW_Execution_Mode_change**REFINES** TC_Execution_Mode_change

ANY
tc_handler
WHERE
 $grd1 : tc_handler \in dom(TCpool)$
 $grd2 : TCstatus(tc_handler) = Waiting_for_Execution$
 $grd3 : Type_of_TC(TCpool(tc_handler)) \in MODE_CHANGING_TC$
THEN
 $act1 : TCstatus(tc_handler) := Successfully_Executed$
END

EVENT Non_TC_Execution_Mode_change

ANY
new_mode
WHERE
 $grd4 : new_mode \in CSW_MODES$
 $grd5 : ((csw_mode \mapsto new_mode) \in CSW_TC_TRANSITIONS) \vee ((csw_mode \mapsto new_mode) \in CSW_FDIR_TRANSITIONS) \vee ((csw_mode \mapsto new_mode) \in CSW_PSEUDO_TC_TRANSITIONS)$
THEN
 $act3 : csw_mode := new_mode$
END

EVENT FDIR_Execution_Mode_change

ANY
new_mode
WHERE
 $grd4 : new_mode \in CSW_MODES$
 $grd5 : ((csw_mode \mapsto new_mode) \in CSW_FDIR_TRANSITIONS) \vee ((csw_mode \mapsto new_mode) \in CSW_PSEUDO_TC_TRANSITIONS)$
THEN
 $act3 : csw_mode := new_mode$
END

EVENT TC_Execution_HK_on**REFINES** TC_Execution_HK_on

ANY
tc_handler
WHERE
 $grd1 : tc_handler \in dom(TCpool)$
 $grd2 : TCstatus(tc_handler) = Waiting_for_Execution$
 $grd3 : Type_of_TC(TCpool(tc_handler)) = HK_on_TC$
THEN
 $act1 : TCstatus(tc_handler) := Successfully_Executed$
 $act2 : csw_hk_on := TRUE$

END

EVENT TC_Execution_HK_off

REFINES TC_Execution_HK_off

ANY

tc_handler

WHERE

grd1 : *tc_handler* \in *dom*(TCpool)

grd2 : TCstatus(*tc_handler*) = *Waiting_for_Execution*

grd3 : Type_of_TC(TCpool(*tc_handler*)) = *HK_off_TC*

THEN

act1 : TCstatus(*tc_handler*) := *Successfully_Executed*

act2 : *csw_hk_on* := *FALSE*

END

EVENT TC_Execution_SCI_on

REFINES TC_Execution_SCI_on

ANY

tc_handler

WHERE

grd1 : *tc_handler* \in *dom*(TCpool)

grd2 : TCstatus(*tc_handler*) = *Waiting_for_Execution*

grd3 : Type_of_TC(TCpool(*tc_handler*)) = *SCI_on_TC*

THEN

act1 : TCstatus(*tc_handler*) := *Successfully_Executed*

act2 : *csw_sci_on* := *TRUE*

END

EVENT TC_Execution_SCI_off

REFINES TC_Execution_SCI_off

ANY

tc_handler

WHERE

grd1 : *tc_handler* \in *dom*(TCpool)

grd2 : TCstatus(*tc_handler*) = *Waiting_for_Execution*

grd3 : Type_of_TC(TCpool(*tc_handler*)) = *SCI_off_TC*

THEN

act1 : TCstatus(*tc_handler*) := *Successfully_Executed*

act2 : *csw_sci_on* := *FALSE*

END

EVENT Reset_Output_Buffer

REFINES Reset_Output_Buffer

ANY

tc_handler

WHERE

grd1 : *tc_handler* \in *dom*(TCpool)

grd2 : TCstatus(*tc_handler*) = *Waiting_for_Execution*

grd3 : Type_of_TC(TCpool(*tc_handler*)) = *TM_Buffer_Reset_TC*

THEN

act1 : *TMbuffer* := \emptyset

act2 : TCstatus(*tc_handler*) := *Successfully_Executed*

END

EVENT TC_Execution_Failure

REFINES TC_Execution_Failure

ANY

tc_handler
WHERE
 $grd1 : tc_handler \in dom(TCpool)$
 $grd2 : TCstatus(tc_handler) = Waiting_for_Execution$
THEN
 $act1 : TCstatus(tc_handler) := Execution_Failed$
END

EVENT Report_TC_execution_success**REFINES** Report_TC_execution_success

ANY
 $tc_handler$
 $tm_handler$
 tm
WHERE
 $grd1 : tc_handler \in dom(TCpool)$
 $grd2 : tm_handler \in TCTM_POOL_ADDRESSES$
 $grd3 : tm \in TM_SET$
 $grd4 : TCstatus(tc_handler) = Successfully_Executed$
 $grd5 : tm_handler \notin dom(TCpool)$
 $grd6 : tm_handler \notin dom(TMpool)$
 $grd7 : Type_of_TM(tm) = Exec_ok_TM$
THEN
 $act1 : TCpool := \{tc_handler\} \triangleleft TCpool$
 $act2 : TCstatus := \{tc_handler\} \triangleleft TCstatus$
 $act3 : TMpool(tm_handler) := tm$
END

EVENT Report_TC_execution_failure**REFINES** Report_TC_execution_failure

ANY
 $tc_handler$
 $tm_handler$
 tm
WHERE
 $grd1 : tc_handler \in dom(TCpool)$
 $grd2 : tm_handler \in TCTM_POOL_ADDRESSES$
 $grd3 : tm \in TM_SET$
 $grd4 : TCstatus(tc_handler) = Execution_Failed$
 $grd5 : tm_handler \notin dom(TCpool)$
 $grd6 : tm_handler \notin dom(TMpool)$
 $grd7 : Type_of_TM(tm) = Exec_fail_TM$
THEN
 $act1 : TCpool := \{tc_handler\} \triangleleft TCpool$
 $act2 : TCstatus := \{tc_handler\} \triangleleft TCstatus$
 $act3 : TMpool(tm_handler) := tm$
END

EVENT Store_HK_Data**REFINES** Store_HK_Data

ANY
 $tm_handler$
 tm
WHERE
 $grd1 : tm_handler \in TCTM_POOL_ADDRESSES$
 $grd2 : tm \in TM_SET$
 $grd3 : tm_handler \notin dom(TCpool)$

```

    grd4 : tm_handler  $\notin$  dom(TMpool)
    grd5 : csw_hk_on = TRUE
    grd6 : Type_of_TM(tm) = HK_data_TM
THEN
    act1 : TMpool(tm_handler) := tm
END

```

EVENT Store_SCI_Data**REFINES** Store_SCI_Data

```

ANY
    tm_handler
    tm
WHERE
    grd1 : tm_handler  $\in$  TCTM_POOL_ADDRESSES
    grd2 : tm  $\in$  TM_SET
    grd3 : tm_handler  $\notin$  dom(TCpool)
    grd4 : tm_handler  $\notin$  dom(TMpool)
    grd5 : csw_sci_on = TRUE
    grd6 : Type_of_TM(tm) = SCI_data_TM
THEN
    act1 : TMpool(tm_handler) := tm
END

```

EVENT Pass_TM**REFINES** Pass_TM

```

ANY
    tm_handler
    tm_pointer
WHERE
    grd1 : tm_handler  $\in$  dom(TMpool)
    grd2 : tm_pointer  $\in$  TM_BUFFER_ADDRESSES
    grd3 : tm_pointer  $\notin$  dom(TMbuffer)
THEN
    act2 : TMbuffer(tm_pointer) := TMpool(tm_handler)
    act1 : TMpool := {tm_handler}  $\triangleleft$  TMpool
END

```

EVENT Drop_HK_TM**REFINES** Drop_HK_TM

```

ANY
    tm_handler
WHERE
    grd1 : tm_handler  $\in$  dom(TMpool)
    grd2 : csw_hk_on = FALSE
    grd3 : Type_of_TM(TMpool(tm_handler)) = HK_data_TM
THEN
    act1 : TMpool := {tm_handler}  $\triangleleft$  TMpool
END

```

EVENT Drop_SCI_TM**REFINES** Drop_SCI_TM

```

ANY
    tm_handler
WHERE
    grd1 : tm_handler  $\in$  dom(TMpool)
    grd2 : csw_sci_on = FALSE
    grd3 : Type_of_TM(TMpool(tm_handler)) = SCI_data_TM

```

THEN

$act1 : TMpool := \{tm_handler\} \triangleleft TMpool$

END

EVENT env_Deliver_TM

REFINES env_Deliver_TM

ANY

$tm_pointer$

WHERE

$grd1 : tm_pointer \in dom(TMbuffer)$

THEN

$act1 : TMbuffer := \{tm_pointer\} \triangleleft TMbuffer$

END

END