

=====
An Event-B Specification of ABS_DISPLAY6 @ Date: 16 Nov 2007 **@ Time:** 08:23:37 PM
=====

MACHINE ABS_DISPLAY6

REFINES ABS_DISPLAY5

SEES CDIS_CONTEXT6

VARIABLES

database
page_selections
pages
previous_pages
private_pages
trq
time_now
edd_acks_required
concealed_displays

INVARIANTS

inv2 : $\forall ai \cdot ai \in Attr_id \Rightarrow ai \mapsto value(database(ai)) \in conform$ /* The purpose of the invariant is to check that the field type conforms to that specified in the initial specification.
*/

EVENTS

INITIALISATION

BEGIN

act3 : *page_selections* := \emptyset
act8 : *database* := *db0*
act9 : *previous_pages* := \emptyset
act10 : *private_pages* := \emptyset
act1 : *pages* := \emptyset
act11 : *trq* := \emptyset
act12 : *time_now* : $\in Date_time$
act4 : *edd_acks_required* := \emptyset
act2 : *concealed_displays* := \emptyset

END

EVENT UPDATE_DATABASE

/* This event has been affected by the introduction of Conform Function. */

REFINES UPDATE_DATABASE

ANY

ups
changed_ais
selected_ais
update_mesg

WHERE

grd2 : *ups* $\in Attr_id \leftrightarrow Attrs$
grd4 : (*ups*; *last_update*) = *dom*(*ups*) \times {*time_now*}
grd5 : (*ups*; *manually_updated*) = *dom*(*ups*) \times {*FALSE*}
grd7 : $\forall ai \cdot ai \in dom(ups) \Rightarrow ai \mapsto value(ups(ai)) \in conform$ /* New guard has been added

to demonstrate the type checking process. */

$grd3 : changed_ais \subseteq Attr_id$

$grd6 : changed_ais = \{ai \mid ai \in dom(ups) \wedge value(database(ai)) \neq value(ups(ai))\}$

$grd1 : selected_ais \in EDD_id \leftrightarrow Attr_id$

$grd8 : selected_ais = page_selections; pages; contents; fields; attr$

$grd9 : update_mesg \in MSG$

$grd10 : update_mesg = Update_successful$ /* New output message has been added to

reflect successful outcome of type checking. */

THEN

$act1 : database := database \Leftarrow ups$

$act2 : edd_acks_required := edd_acks_required \cup ((selected_ais \triangleright changed_ais) \triangleright Acknowledge)$

END

EVENT UPDATE_DATABASE_FAILED

/* This is a new event has been introduced to as a aresult of type checking and the introduction of Conform Function. */

ANY

ups

$update_mesg$

WHERE

$grd2 : ups \in Attr_id \leftrightarrow Attrs$

$grd7 : \exists ai. ai \in dom(ups) \Rightarrow ai \mapsto value(ups(ai)) \notin conform$ /* New guard has been added

to demonstrate the type checking process. */

$grd9 : update_mesg \in MSG$

$grd10 : update_mesg = Update_failed$ /* New output message has been added to reflect

unsuccessful outcome of type checking. */

THEN

$skip$

END

EVENT SET_DATA_VALUE

/* This event has been affected by the introduction of Conform Function. */

REFINES SET_DATA_VALUE

ANY

ei

ai

av

a

$selected_ais$

set_data_mesg

WHERE

$grd1 : ei \in EDD_id$

$grd4 : ei \in EDITORS$

$grd2 : ai \in Attr_id$

$grd3 : av \in Attr_value$

$grd5 : a \in Attrs$

$grd11 : ai \mapsto av \in conform$ /* New guard has been added to demonstrate the type checking

process. */

$grd6 : value(a) = av$

$grd7 : last_update(a) = time_now$

$grd10 : manually_updated(a) = TRUE$

$grd9 : selected_ais \in EDD_id \leftrightarrow Attr_id$

$grd8 : selected_ais = page_selections; pages; contents; fields; attr$

$grd12 : set_data_mesg \in MSG$

$grd13 : set_data_mesg = Update_successful$ /* New output message has been added to

reflect successful outcome of type checking. */

THEN

```

    act1 : database(ai) := a
    act2 : edd_acks_required := edd_acks_required  $\cup$  ((selected_ais  $\triangleright$  {ai})  $\triangleright$  Acknowledge)
END

```

EVENT SET_DATA_VALUE_FAILED

/* This is a new event has been introduced to as a aresult of type checking and the introduction of Conform Function. */

```

ANY
    ei
    ai
    av
    set_data_mesg
WHERE
    grd1 : ei  $\in$  EDD_id
    grd4 : ei  $\in$  EDITORS
    grd2 : ai  $\in$  Attr_id
    grd3 : av  $\in$  Attr_value
    grd11 : ai  $\mapsto$  av  $\notin$  conform /* New guard has been added to demonstrate the type checking
process. */
    grd12 : set_data_mesg  $\in$  MSG
    grd13 : set_data_mesg = Update_failed /* New output message has been added to reflect
unsuccessful outcome of type checking. */
THEN
    skip
END

```

EVENT DISPLAY_PAGE

REFINES DISPLAY_PAGE

```

ANY
    ei
    no
WHERE
    grd1 : ei  $\in$  EDD_id
    grd2 : no  $\in$  Page_number
    grd3 : no  $\in$  dom(pages)
THEN
    act1 : page_selections(ei) := no
    act2 : edd_acks_required := {ei}  $\triangleleft$  edd_acks_required
END

```

EVENT DISMISS_PAGE

REFINES DISMISS_PAGE

```

ANY
    ei
WHERE
    grd1 : ei  $\in$  EDD_id
    grd2 : ei  $\in$  dom(page_selections)
THEN
    act2 : page_selections := {ei}  $\triangleleft$  page_selections
    act1 : edd_acks_required := {ei}  $\triangleleft$  edd_acks_required
    act3 : concealed_displays := concealed_displays  $\setminus$  {ei}
END

```

EVENT ADD_PAGE

REFINES ADD_PAGE

```

ANY
    ei

```

no
pc
p
WHERE
grd1 : $ei \in EDD_id$
grd2 : $no \in Page_number$
grd3 : $pc \in Page_contents$
grd4 : $ei \in EDITORS$
grd5 : $p \in Page$
grd6 : $contents(p) = pc$
grd7 : $creation_date(p) = time_now$
THEN
act1 : $private_pages(no) := p$
END

EVENT RELEASE_PAGE
REFINES RELEASE_PAGE

ANY
no
selected_ais
pc1
pc2
WHERE
grd1 : $no \in Page_number$
grd2 : $no \in dom(private_pages)$
grd3 : $no \in dom(pages)$
grd5 : $pc1 = contents(pages(no))$
grd6 : $pc2 = contents(private_pages(no))$
grd4 : $(pc1 \in Overlay_Page_contents \Rightarrow pc2 \in Overlay_Page_contents)$
grd8 : $selected_ais \in EDD_id \leftrightarrow Attr_id$
grd7 : $selected_ais = (page_selections \triangleright \{no\}); private_pages; contents; fields; attr$
THEN
act1 : $pages(no) := private_pages(no)$
act3 : $previous_pages(no) := pages(no)$
act2 : $private_pages := \{no\} \triangleleft private_pages$
act5 : $edd_acks_required := edd_acks_required \triangleleft ((selected_ais \triangleright Acknowledge) \cap edd_acks_required)$
END

EVENT RELEASE_PAGE_CURRENT
REFINES RELEASE_PAGE_CURRENT

ANY
no
selected_ais
WHERE
grd1 : $no \in Page_number$
grd2 : $no \in dom(private_pages)$
grd3 : $no \notin dom(pages)$
grd5 : $selected_ais \in EDD_id \leftrightarrow Attr_id$
grd4 : $selected_ais = (page_selections \triangleright \{no\}); private_pages; contents; fields; attr$
THEN
act1 : $pages(no) := private_pages(no)$
act2 : $private_pages := \{no\} \triangleleft private_pages$
act3 : $edd_acks_required := edd_acks_required \triangleleft ((selected_ais \triangleright Acknowledge) \cap edd_acks_required)$
END

EVENT RELEASE_PAGE_OVERLAY
REFINES RELEASE_PAGE_OVERLAY

ANY

no
selected_ais
edd_ids

WHERE

grd1 : *no* \in *Page_number*
grd2 : *no* \in *dom(private_pages)*
grd3 : *no* \in *dom(pages)*
grd6 : *contents(pages(no))* \in *Overlay_Page_contents*
grd7 : *contents(private_pages(no))* \notin *Overlay_Page_contents*
grd5 : *selected_ais* \in *EDD_id* \leftrightarrow *Attr_id*
grd4 : *selected_ais* = (*page_selections* \triangleright {*no*}); *private_pages*; *contents*; *fields*; *attr*
grd8 : *edd_ids* = *dom(page_selections* \triangleright {*no*})

THEN

act1 : *pages(no)* := *private_pages(no)*
act3 : *previous_pages(no)* := *pages(no)*
act2 : *private_pages* := {*no*} \triangleleft *private_pages*
act4 : *edd_acks_required* := *edd_acks_required* \triangleleft ((*selected_ais* \triangleright *Acknowledge*) \cap *edd_acks_required*)
act5 : *concealed_displays* := *concealed_displays* \setminus *edd_ids*

END**EVENT RELEASE_PAGES_FROM_TRQ****REFINES RELEASE_PAGES_FROM_TRQ****ANY**

ss
selected_ais
pp
pn
p_nums
edd_ids

WHERE

grd1 : *ss* \in *Page_number* \leftrightarrow *Page*
grd3 : *pp* = {*rp* | *rp* \in *Rel_Page* \wedge (*release_date(rp)* \mapsto *time_now*) \in *leq*}
grd2 : *ss* = *trq* \triangleright *pp*
grd5 : *selected_ais* \in *EDD_id* \leftrightarrow *Attr_id*
grd4 : *selected_ais* = *page_selections*; *ss*; *contents*; *fields*; *attr*
grd6 : *pn* = *dom(ss)*
grd7 : *p_nums* = *dom(((pages; contents) \triangleright *Overlay_Page_contents*) \cap ((*ss*; *contents*) \triangleright *Overlay_Page_contents*))*
grd8 : *edd_ids* = *dom(page_selections* \triangleright *p_nums*)

THEN

act1 : *pages* := *pages* \triangleleft *ss*
act3 : *previous_pages* := *previous_pages* \triangleleft (*pn* \triangleleft *pages*)
act2 : *trq* := *trq* \setminus *ss*
act4 : *edd_acks_required* := *edd_acks_required* \triangleleft ((*selected_ais* \triangleright *Acknowledge*) \cap *edd_acks_required*)
act5 : *concealed_displays* := *concealed_displays* \setminus *edd_ids*

END**EVENT DELETE_PAGE****REFINES DELETE_PAGE****ANY**

ei
no
edd_ids

WHERE

grd1 : *ei* \in *EDD_id*
grd2 : *no* \in *Page_number*
grd3 : *ei* \in *EDITORS*

```

    grd4 : edd_ids = dom(page_selections  $\triangleright$  {no})
  THEN
    act1 : pages := {no}  $\Leftarrow$  pages
    act5 : previous_pages := {no}  $\Leftarrow$  previous_pages
    act2 : private_pages := {no}  $\Leftarrow$  private_pages
    act3 : trq := {no}  $\Leftarrow$  trq
    act4 : page_selections := page_selections  $\triangleright$  {no}
    act6 : edd_acks_required := dom(page_selections  $\triangleright$  {no})  $\triangleleft$  edd_acks_required
    act7 : concealed_displays := concealed_displays \ edd_ids
  END

```

EVENT ADD_PAGE_TO_TRQ**REFINES** ADD_PAGE_TO_TRQ

```

  ANY
    ei
    no
    pc
    p
  WHERE
    grd1 : ei  $\in$  EDD_id
    grd2 : no  $\in$  Page_number
    grd3 : pc  $\in$  Page_contents
    grd4 : ei  $\in$  EDITORS
    grd5 : p  $\in$  Rel_Page
    grd6 : contents(p) = pc
    grd7 : creation_date(p) = time_now
    grd8 : (time_now  $\mapsto$  release_date(p))  $\in$  leg
  THEN
    act1 : trq(no) := p
  END

```

EVENT CLOCK**REFINES** CLOCK

```

  ANY
    time_next
  WHERE
    grd1 : time_next  $\in$  Date_time
    grd2 : time_now  $\mapsto$  time_next  $\in$  leg
    grd3 : time_next  $\neq$  time_now
  THEN
    act1 : time_now := time_next
  END

```

EVENT ACKNOWLEDGE_DATA**REFINES** ACKNOWLEDGE_DATA

```

  ANY
    ei
  WHERE
    grd1 : ei  $\in$  EDD_id
    grd2 : ei  $\in$  dom(page_selections)
  THEN
    act1 : edd_acks_required := {ei}  $\Leftarrow$  edd_acks_required
  END

```

EVENT VIEW_PAGE**REFINES** VIEW_PAGE

```

  ANY

```

```

    ei
    dp
    ed
WHERE
    grd1 : ei ∈ dom(page_selections)
    grd2 : dp ∈ Disp_params
    grd3 : ed ∈ EDD_display
    grd4 : dp_data(dp) = database
    grd5 : dp_page(dp) = pages(page_selections(ei))
    grd7 : dp_time(dp) = time_now
    grd6 : ed = disp_values(dp)
    grd8 : dp_acks_required(dp) = edd_acks_required{ei}
    grd9 : dp_concealed(dp) = bool(ei ∈ concealed_displays)
THEN
    skip
END

EVENT TOGGLE_REVEAL_CONCEAL1
REFINES TOGGLE_REVEAL_CONCEAL1
ANY
    ei
WHERE
    grd11 : ei ∈ EDD_id
    grd12 : ei ∈ concealed_displays
THEN
    act11 : concealed_displays := concealed_displays \ {ei}
END

EVENT TOGGLE_REVEAL_CONCEAL2
REFINES TOGGLE_REVEAL_CONCEAL2
ANY
    ei
WHERE
    grd1 : ei ∈ EDD_id
    grd2 : ei ∈ dom(page_selections)
    grd3 : ei ∉ concealed_displays
    grd4 : contents(pages(page_selections(ei))) ∈ Overlay_Page_contents
THEN
    act1 : concealed_displays := concealed_displays ∪ {ei}
END

END

```