

=====
An Event-B Specification of ABS_DISPLAY3 @ Date: 16 Nov 2007 **@ Time:** 05:36:19 PM
=====

MACHINE ABS_DISPLAY3

REFINES ABS_DISPLAY2

SEES CDIS_CONTEXT3

VARIABLES

database
page_selections
pages
previous_pages
private_pages
trq
time_now
edd_acks_required /* Defining critical subset of attributes that they need operators acknowledgment after any update. *edd_acks_required* models the sets of airport data displayed on EDDs for which acknowledgments of highlighted data are outstanding. When airport data requiring acknowledgment changes, the data items are added to the sets associated with EDDs displaying the data. If a data item changes again before an acknowledgment is given, the fields displaying the new value remain highlighted. Once acknowledged, the data is unhighlighted and removed from the *edd_acks_required* state. */

INVARIANTS

inv2 : *edd_acks_required* \in *EDD_id* \leftrightarrow *Attr_id*
inv1 : *ran*(*edd_acks_required*) \subseteq *Acknowledge*

EVENTS

INITIALISATION

BEGIN

act3 : *page_selections* := \emptyset
act8 : *database* : \in *Attr_id* \rightarrow *Attrs*
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

END

EVENT UPDATE_DATABASE

/* This event has been affected by the use of *edd_acks_required*. */

REFINES UPDATE_DATABASE

ANY

ups
changed_ais
selected_ais

WHERE

grd2 : *ups* \in *Attr_id* \leftrightarrow *Attrs*
grd4 : (*ups*; *last_update*) = *dom*(*ups*) \times {*time_now*}

$grd3 : changed_ais \subseteq Attr_id$
 $grd6 : changed_ais = \{ai \mid ai \in dom(ups) \wedge value(database(ai)) \neq value(ups(ai))\}$ /* Auxiliary variable introduced to assist with updating list of critical fields which they need acknowledgment. Restricting the update action to the attributes which actually have been changed. */
 $grd1 : selected_ais \in EDD_id \leftrightarrow Attr_id$
 $grd8 : selected_ais = page_selections; pages; contents; fields; attr$ /* Auxiliary variable introduced to assist with updating list of critical fields which they need acknowledgment. */
THEN
 $act1 : database := database \Leftarrow ups$
 $act2 : edd_acks_required := edd_acks_required \cup ((selected_ais \triangleright changed_ais) \triangleright Acknowledge)$
/* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When a critical attribute has been updated automatically it should be added to the list of field which they need acknowledgment. */
END

EVENT SET_DATA_VALUE

/* This event has been affected by the use of edd_acks_required. */

REFINES SET_DATA_VALUE

ANY
 ei
 ai
 av
 a
 $selected_ais$
WHERE
 $grd1 : ei \in EDD_id$
 $grd4 : ei \in EDITORS$
 $grd2 : ai \in Attr_id$
 $grd3 : av \in Attr_value$
 $grd5 : a \in Attrs$
 $grd6 : value(a) = av$
 $grd7 : last_update(a) = time_now$
 $grd9 : selected_ais \in EDD_id \leftrightarrow Attr_id$
 $grd8 : selected_ais = page_selections; pages; contents; fields; attr$ /* Auxiliary variable introduced to assist with updating list of critical fields which they need acknowledgment. */
THEN
 $act1 : database(ai) := a$
 $act2 : edd_acks_required := edd_acks_required \cup ((selected_ais \triangleright \{ai\}) \triangleright Acknowledge)$ /* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When an attribute has been updated manually it should be added to the list of field which they need acknowledgment. */
END

EVENT DISPLAY_PAGE

/* This event has been affected by the use of edd_acks_required. */

REFINES DISPLAY_PAGE

ANY
 ei
 no
WHERE
 $grd1 : ei \in EDD_id$
 $grd2 : no \in Page_number$
 $grd3 : no \in dom(pages)$
THEN
 $act1 : page_selections(ei) := no$
 $act2 : edd_acks_required := \{ei\} \Leftarrow edd_acks_required$ /* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When a new page has

been displayed for the first time, the specification has assumed that its should not have any critical fields to be added to the list of field which they need acknowledgement. */

END

EVENT DISMISS_PAGE

/* This event has been affected by the use of edd_acks_required. */

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 : $edd_acks_required := \{ei\} \triangleleft edd_acks_required$ /* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When a new page has been deleted its critical fields should be removed from the list of field which they need acknowledgement. */

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

/* This event has been affected by the use of edd_acks_required. */

REFINES RELEASE_PAGE

ANY

no

selected_ais

WHERE

grd1 : $no \in Page_number$

grd2 : $no \in dom(private_pages)$

grd3 : $no \in dom(pages)$

grd8 : $selected_ais \in EDD_id \leftrightarrow Attr_id$

grd7 : $selected_ais = (page_selections \triangleright \{no\}); private_pages; contents; fields; attr$ /* Auxiliary variable introduced to assist with updating list of critical fields which they need acknowledgment. */

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 $\Leftarrow ((selected_ais \triangleright Acknowledge) \cap edd_acks_required)$
 /* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When a new page has been released its critical fields should be added to the list of field which they need acknowledgement. */

END

EVENT RELEASE_PAGE_CURRENT

/* This event has been affected by the use of edd_acks_required. */

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 /* Auxiliary variable introduced to assist with updating list of critical fields which they need acknowledgment. */

THEN

act1 : pages(no) := private_pages(no)

act2 : private_pages := {no} \Leftarrow private_pages

act3 : edd_acks_required := edd_acks_required $\Leftarrow ((selected_ais \triangleright Acknowledge) \cap edd_acks_required)$

/* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When a new page has been released its critical fields should be added to the list of fields which they need acknowledgement. */

END

EVENT RELEASE_PAGES_FROM_TRQ

/* This event has been affected by the use of edd_acks_required. */

REFINES RELEASE_PAGES_FROM_TRQ

ANY

ss

pp

pn

selected_ais

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 /* Auxiliary variable introduced to assist with updating list of critical fields which they need acknowledgment. */

grd6 : pn = dom(ss)

THEN

act1 : pages := pages \Leftarrow ss

act3 : previous_pages := previous_pages \Leftarrow (pn \triangleleft pages)

act2 : trq := trq \setminus ss

act4 : edd_acks_required := edd_acks_required $\Leftarrow ((selected_ais \triangleright Acknowledge) \cap edd_acks_required)$

/* Refinement which triggered by the introduction of critical fields and the need for operator acknowledgment. When a new page has been released its critical fields should be added to the list of field which they need acknowledgement. */

END

EVENT DELETE_PAGE

/* This event has been affected by the use of edd_acks_required. */

REFINES DELETE_PAGE**ANY***ei**no***WHERE***grd1* : *ei* ∈ *EDD_id**grd2* : *no* ∈ *Page_number**grd3* : *ei* ∈ *EDITORS***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* /* When

a page been removed the system has to remove its critical fields from the list of fields which need acknowledgement. */

END**EVENT ADD_PAGE_TO_TRQ****REFINES ADD_PAGE_TO_TRQ****ANY***ei**no**pc**p***WHERE***grd1* : *ei* ∈ *EDD_id**grd2* : *no* ∈ *Page_number**grd3* : *pc* ∈ *Page_contents**grd4* : *ei* ∈ *EDITORS**grd5* : *p* ∈ *Rel_Page**grd6* : *contents*(*p*) = *pc**grd7* : *creation_date*(*p*) = *time_now**grd8* : (*time_now* \mapsto *release_date*(*p*)) ∈ *leq***THEN***act1* : *trq*(*no*) := *p***END****EVENT CLOCK****REFINES CLOCK****ANY***time_next***WHERE***grd1* : *time_next* ∈ *Date_time**grd2* : *time_now* \mapsto *time_next* ∈ *leq**grd3* : *time_next* \neq *time_now***THEN***act1* : *time_now* := *time_next***END****EVENT ACKNOWLEDGE_DATA**/* New event introduced in this level in conjunction with *edd_acks_required* variable. */**ANY***ei***WHERE***grd1* : *ei* ∈ *EDD_id*

```

    grd2 : ei ∈ dom(page_selections)
  THEN
    act1 : edd_acks_required := {ei} ⋈ edd_acks_required    /* Refinement which triggered by
the introduction of critical fields and the need for operator acknowledgment. In this guard we remove
the EDD which its user has acknowledge the critical fields. Acknowledge with take place once for all
critical fields. */
  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}    /* New guard add in conjunction
with edd_acks_required. */
  THEN
    skip
  END

END

```