

=====
An Event-B Specification of ABS_DISPLAY1 @ Date: 14 Nov 2007 @ Time: 04:18:22 PM
=====

MACHINE ABS_DISPLAY1

REFINES ABS_DISPLAY

SEES CDIS_CONTEXT

VARIABLES

database /* CDIS Central Database */
page_selections
pages
private_pages
trq
previous_pages /* A new variable which has been added to hold one- step history of an updated
page layout. */

INVARIANTS

inv2 : *previous_pages* \in *Page_number* \leftrightarrow *Page*
inv4 : *dom(previous_pages)* \subseteq *dom(pages)*

EVENTS

INITIALISATION

BEGIN

act1 : *database* \in *Attr_id* \rightarrow *Attrs*
act3 : *page_selections* := \emptyset
act4 : *private_pages* := \emptyset
act5 : *trq* := \emptyset
act6 : *pages* := \emptyset
act7 : *previous_pages* := \emptyset

END

EVENT UPDATE_DATABASE

REFINES UPDATE_DATABASE

ANY

ups

WHERE

grd1 : *ups* \in *Attr_id* \leftrightarrow *Attrs*

THEN

act1 : *database* := *database* \Leftarrow *ups*

END

EVENT SET_DATA_VALUE

REFINES SET_DATA_VALUE

ANY

ei

ai

av

a

WHERE

grd4 : *ei* \in *EDITORS*

```

    grd2 : ai ∈ Attr_id
    grd3 : av ∈ Attr_value
    grd5 : a ∈ Attrs
    grd6 : value(a) = av
THEN
    act1 : database(ai) := a
END

```

EVENT DISPLAY_PAGE
REFINES DISPLAY_PAGE

```

ANY
    ei
    no
WHERE
    grd1 : ei ∈ EDD_id
    grd2 : no ∈ Page_number
    grd3 : no ∈ dom(pages)
THEN
    act1 : page_selections(ei) := no
END

```

EVENT DISMISS_PAGE
REFINES DISMISS_PAGE

```

ANY
    ei
WHERE
    grd1 : ei ∈ EDD_id
    grd2 : ei ∈ dom(page_selections)
THEN
    act1 : page_selections := {ei} ⧸ page_selections
END

```

EVENT ADD_PAGE
REFINES ADD_PAGE

```

ANY
    ei
    no
    pc
    p
WHERE
    grd1 : ei ∈ EDD_id
    grd2 : no ∈ Page_number
    grd3 : pc ∈ Page_contents
    grd4 : ei ∈ EDITORS
    grd5 : p ∈ Page
    grd6 : contents(p) = pc
THEN
    act1 : private_pages(no) := p
END

```

EVENT RELEASE_PAGE

/* This event has been affected by the introduction of the new history variable *previous_pages* - -
 In this event we update a currently available page layout, hence we need to store the previous page
 layout in *previous_pages*. */

REFINES RELEASE_PAGE

```

ANY
    no

```

WHERE

$grd1 : no \in Page_number$
 $grd2 : no \in dom(private_pages)$
 $grd3 : no \in dom(pages)$

THEN

$act1 : pages(no) := private_pages(no)$
 $act3 : previous_pages(no) := pages(no)$
 $act2 : private_pages := \{no\} \triangleleft private_pages$

END**EVENT RELEASE_PAGE_CURRENT**

/ Releasing a new page layout. In this case there is no page layout history to be stored in the previous_pages. */*

REFINES RELEASE_PAGE**ANY**

no

WHERE

$grd1 : no \in Page_number$
 $grd2 : no \in dom(private_pages)$
 $grd3 : no \notin dom(pages)$

THEN

$act1 : pages(no) := private_pages(no)$
 $act2 : private_pages := \{no\} \triangleleft private_pages$

END**EVENT RELEASE_PAGES_FROM_TRQ**

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

REFINES RELEASE_PAGES_FROM_TRQ**ANY**

ss

WHERE

$grd2 : ss \subseteq trq$

THEN

$act1 : pages := pages \triangleleft ss$
 $act3 : previous_pages := previous_pages \triangleleft (dom(ss) \triangleleft pages)$ */* If any previously available page layout has changed, it has to be added to the previous_pages. */*
 $act2 : trq := trq \setminus ss$

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

ei

no */* This event has been affected by the introduction of previous_pages. */*

WHERE

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

THEN

$act1 : pages := \{no\} \triangleleft pages$
 $act5 : previous_pages := \{no\} \triangleleft previous_pages$ */* If a page layout has been deleted then its history should be deleted as well. */*
 $act2 : private_pages := \{no\} \triangleleft private_pages$
 $act3 : trq := \{no\} \triangleleft trq$
 $act4 : page_selections := page_selections \triangleright \{no\}$

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 : $p \in Page$

grd6 : $contents(p) = pc$

THEN

act1 : $trq(no) := p$

END

EVENT VIEW_PAGE

REFINES VIEW_PAGE

ANY

ei

dp

ed

WHERE

grd1 : $ei \in dom(page_selections)$

grd2 : $dp \in Disp_params$

grd3 : $ed \in EDD_display$

grd4 : $dp_data(dp) = database$

grd5 : $dp_page(dp) = pages(page_selections(ei))$

grd6 : $ed = disp_values(dp)$

THEN

skip

END

END