MODULE *CompositionalSpec*

EXTENDS *Naturals*, *Sequences*, *TLC*

CONSTANT *Apps*, *Permissions*

EnvironmentId $\triangleq$ "EnvironmentId"
ApsId $\triangleq$ "ApsId"
MaxAllowedUpdates $\triangleq$ 2

**CONSTANTS **
Null $\triangleq$ "NULL"
PERMISSION_DENIED $\triangleq$ "PERMISSION_DENIED"
PERMISSION_ALLOWED $\triangleq$ "PERMISSION_ALLOWED"
PERMISSION_ONLY_ONCE $\triangleq$ "PERMISSION_ONLY_ONCE"
PERMISSION_WHILE_USING_APP $\triangleq$ "PERMISSION_WHILE_USING_APP"
ACTION_RECORD_AUDIO $\triangleq$ "ACTION_RECORD_AUDIO"
DATUM_SYSTEM_STATE $\triangleq$ "DATUM_SYSTEM_STATE"
PERMISSION_TYPE_URI $\triangleq$ "PERMISSION_TYPE_URI"
PERMISSION_TYPE_CUSTOM $\triangleq$ "PERMISSION_TYPE_CUSTOM"
PERMISSION_TYPE_NORMAL $\triangleq$ "PERMISSION_TYPE_NORMAL"
PERMISSION_TYPE_RUNTIME $\triangleq$ "PERMISSION_TYPE_RUNTIME"
PERMISSION_TYPE_SPECIAL $\triangleq$ "PERMISSION_TYPE_SPECIAL"
PERMISSION_TYPE_SIGNATURE $\triangleq$ "PERMISSION_TYPE_SIGNATURE"
PROTECTION_LEVEL_NORMAL $\triangleq$ "PROTECTION_LEVEL_NORMAL"
PROTECTION_LEVEL_APP_OP $\triangleq$ "PROTECTION_LEVEL_APP_OP"
PROTECTION_LEVEL_SIGNATURE $\triangleq$ "PROTECTION_LEVEL_SIGNATURE"
PROTECTION_LEVEL_DANGEROUS $\triangleq$ "PROTECTION_LEVEL_DANGEROUS"
**END* OF CONSTANTS **

**ENUMS **
Boolean $\triangleq$ {TRUE, FALSE, *Null*}
DatumType $\triangleq$ {*DATUM_SYSTEM_STATE*}
ActionType $\triangleq$ {*ACTION_RECORD_AUDIO*}
ConsentType $\triangleq$ {*PERMISSION_DENIED*, *PERMISSION_ALLOWED*,
PERMISSION_ONLY_ONCE, *PERMISSION_WHILE_USING_APP*}
ProtectionLevel $\triangleq$ {*PROTECTION_LEVEL_NORMAL*, *PROTECTION_LEVEL_SIGNATURE*,
PROTECTION_LEVEL_DANGEROUS, *PROTECTION_LEVEL_APP_OP*}
PermissionType $\triangleq$ {*PERMISSION_TYPE_NORMAL*, *PERMISSION_TYPE_SIGNATURE*,
PERMISSION_TYPE_RUNTIME, *PERMISSION_TYPE_SPECIAL*,
PERMISSION_TYPE_URI, *PERMISSION_TYPE_CUSTOM*}
**END* OF *ENUMS* **

** **--algorithm** *Universe*
{
variables *env_vars* =
[*actions* $\mapsto$ {}, *permission_groups* $\mapsto$ {},

```
 applications ↦
  [a ∈ Apps  ↦ [installed ↦ FALSE, version ↦ 0, terminated ↦ FALSE]],
   data ↦ {}, permissions ↦ [p ∈ Permissions ↦ [a ∈ Apps ↦ Null]]] ;

app_vars = [a ∈ Apps ↦
             [manifest ↦ [p ∈ Permissions ↦ Null],
              signature ↦ {}, private_keys ↦ {}, public_key ↦ {},
              certificate ↦ {}, package ↦ {}, services ↦ {},
              receivers ↦ {}, activities ↦ {}, content_providers ↦ {}]] ;

aps_vars = [permission_history ↦ {},
            app_permissions ↦ [a ∈ Apps ↦ [p ∈ Permissions ↦ Null]]] ;

permission_type = Null

procedure installApp(app){INSTALL_APP: env_vars.applications[app].installed := TRUE ; return ; }
procedure uninstallApp(app){UNINSTALL_APP: env_vars.applications[app].installed := FALSE ; retur
procedure updateApp(app){UPDATE_APP: env_vars.applications[app].version := env_vars.applications
procedure terminate(app){TERMINATED: env_vars.applications[self].terminated := TRUE ; return ; }
procedure declarePermission(app, perm){DECLARE_PERMISSION: app_vars[app].manifest[perm] :=
procedure revokePermission(app){REVOKE_PERMISSION: aps_vars.app_permissions[app] := [p ∈ Pe
procedure requestPermission(app, perm)
{
 REQUEST_PERMISSION:
 if (aps_vars.app_permissions[app][perm] ≠ Null)
    return ;

 SWITCH_PERMISSION:
 with (x ∈ PermissionType)
 {
  permission_type := x ;

  if (permission_type = PERMISSION_TYPE_NORMAL ∨ permission_type = PERMISSION_TYPE_SI
      ∨ permission_type = PERMISSION_TYPE_SPECIAL)
  {
    out of scope
   skip ;
  }
  else if (permission_type = PERMISSION_TYPE_RUNTIME ∨ permission_type = PERMISSION_TY
      ∨ permission_type = PERMISSION_TYPE_CUSTOM)
  {
   with (r ∈ {TRUE, FALSE})
   {
    aps_vars.app_permissions[app][perm] := Null ;
   }
  }
 } ;
```

```
 return;
}
 procedure grantUriPermission(app){GRANT_URI_PERMISSION : return;}
 procedure revokeUriPermission(app){REVOKE_URI_PERMISSION : return;}
 procedure checkUriPermission(app){CHECK_URI_PERMISSION : return;}
 procedure checkSelfPermission(app){CHECK_SELF_PERMISSION : return;}
 procedure shouldShowRequestPermissionRationale(app){SHOULD_SHOW_REQUEST_PERMISSION_RATIONALE : re
 procedure requestMultiplePermissions(app){REQUEST_MULTIPLE_PERMISSIONS : return;}
 procedure removeUnusedPermissions(app){REMOVE_UNUSED_PERMISSIONS : return;}

fair process (EnvNext = EnvironmentId)
    variables applications;
{
    EnvBegin:- while ( TRUE )
    {
        either { skip; }
        or
        {
          with ( a ∈ Apps )
          {
           call revokePermission(a);
          }
        }
    } ;
}

fair process ( AppNext ∈ Apps )
    variables i;
{
    AppBegin:- while ( env_vars.applications[self].terminated ≠ TRUE )
    {
     either
     {
      call terminate(self);
     }
     or
     {
      if ( env_vars.applications[self].installed = TRUE )
      {
       either
       {
         call uninstallApp(self);
       }
       or
       {
```

```
            if ( env_vars.applications[self].version < MaxAllowedUpdates )
            {
             call updateApp(self);
            }
           }
          }
          else
          {
           DECLARING_PERMISSION: with ( p ∈ Permissions ) { call declarePermission(self, p); } ;
           INSTALLING_APP: call installApp(self);
          }
         } ;
    }

   fair process ( ApsNext = ApsId )
       variables i;
   {
       ApsBegin:- while ( TRUE )
       {
        either { skip; }
        or
        {
         with ( a ∈ Apps )
         {
          with ( p ∈ Permissions )
          {
           call requestPermission(a, p);
          }
         }
        }
       } ;
    }
}
```

**

BEGIN TRANSLATION ($chksum(pcal) =$ "8e0c5402" $\wedge\ chksum(tla) =$ "9d9aef41")

Process variable i of process *AppNext* at line 121 col 19 changed to $i\_$

Parameter *app* of procedure *installApp* at line 62 col 26 changed to *app_*

Parameter *app* of procedure *uninstallApp* at line 63 col 28 changed to *app_u*

Parameter *app* of procedure *updateApp* at line 64 col 25 changed to *app_up*

Parameter *app* of procedure terminate at line 65 col 25 changed to *app_t*

Parameter *app* of procedure *declarePermission* at line 66 col 33 changed to *app_d*

Parameter *perm* of procedure *declarePermission* at line 66 col 38 changed to *perm_*

Parameter *app* of procedure *revokePermission* at line 67 col 32 changed to *app_r*

CONSTANT *defaultInitValue*

VARIABLES env_vars, app_vars, aps_vars, $permission\_type$, pc, $stack$, $app\_$,
app_u, app_up, app_t, app_d, $perm\_$, app_r, app, $perm$, $applications$,
$i\_$, i

$vars \triangleq \langle env\_vars,\ app\_vars,\ aps\_vars,\ permission\_type,\ pc,\ stack,\ app\_,$
$app\_u,\ app\_up,\ app\_t,\ app\_d,\ perm\_,\ app\_r,\ app,\ perm,\ applications,$
$i\_,\ i\rangle$

$ProcSet \triangleq \{EnvironmentId\} \cup (Apps) \cup \{ApsId\}$

$Init \triangleq$ Global variables
$\land\ env\_vars = [actions \mapsto \{\},\ permission\_groups \mapsto \{\},$
$applications \mapsto$
$[a \in Apps \mapsto [installed \mapsto \text{FALSE},\ version \mapsto 0,\ terminated \mapsto \text{FALSE}]],$
$data \mapsto \{\},\ permissions \mapsto [p \in Permissions \mapsto [a \in Apps \mapsto Null]]]$
$\land\ app\_vars = [a \in Apps \mapsto$
$[manifest \mapsto [p \in Permissions \mapsto Null],$
$signature \mapsto \{\},\ private\_keys \mapsto \{\},\ public\_key \mapsto \{\},$
$certificate \mapsto \{\},\ package \mapsto \{\},\ services \mapsto \{\},$
$receivers \mapsto \{\},\ activities \mapsto \{\},\ content\_providers \mapsto \{\}]]$
$\land\ aps\_vars = [permission\_history \mapsto \{\},$
$app\_permissions \mapsto [a \in Apps \mapsto [p \in Permissions \mapsto Null]]]$
$\land\ permission\_type = Null$
Procedure *installApp*
$\land\ app\_ = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *uninstallApp*
$\land\ app\_u = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *updateApp*
$\land\ app\_up = [self \in ProcSet \mapsto defaultInitValue]$
Procedure terminate
$\land\ app\_t = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *declarePermission*
$\land\ app\_d = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ perm\_ = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *revokePermission*
$\land\ app\_r = [self \in ProcSet \mapsto defaultInitValue]$
Procedure *requestPermission*
$\land\ app = [self \in ProcSet \mapsto defaultInitValue]$
$\land\ perm = [self \in ProcSet \mapsto defaultInitValue]$
Process *EnvNext*
$\land\ applications = defaultInitValue$
Process *AppNext*
$\land\ i\_ = [self \in Apps \mapsto defaultInitValue]$
Process *ApsNext*
$\land\ i = defaultInitValue$
$\land\ stack = [self \in ProcSet \mapsto \langle\rangle]$

$$\begin{array}{l} \wedge\ pc = [self \in ProcSet \mapsto \text{CASE } self = EnvironmentId \rightarrow \text{“EnvBegin”} \\ \qquad\qquad\qquad\qquad\qquad\quad \square\ \ self \in Apps \rightarrow \text{“AppBegin”} \\ \qquad\qquad\qquad\qquad\qquad\quad \square\ \ self = ApsId \rightarrow \text{“ApsBegin”}] \end{array}$$

$$\begin{array}{l} INSTALL\_APP(self) \triangleq \ \wedge\ pc[self] = \text{“INSTALL\_APP”} \\ \qquad\qquad\qquad\qquad\quad \wedge\ env\_vars' = [env\_vars \text{ EXCEPT } !.applications[app\_[self]].installed = \text{TRUE}] \\ \qquad\qquad\qquad\qquad\quad \wedge\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc] \\ \qquad\qquad\qquad\qquad\quad \wedge\ app\_' = [app\_ \text{ EXCEPT } ![self] = Head(stack[self]).app\_] \\ \qquad\qquad\qquad\qquad\quad \wedge\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])] \\ \qquad\qquad\qquad\qquad\quad \wedge \text{ UNCHANGED } \langle app\_vars,\ aps\_vars,\ permission\_type, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad app\_u,\ app\_up,\ app\_t,\ app\_d,\ perm\_,\ app\_r, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad app,\ perm,\ applications,\ i\_,\ i\rangle \end{array}$$

$$installApp(self) \triangleq INSTALL\_APP(self)$$

$$\begin{array}{l} UNINSTALL\_APP(self) \triangleq \ \wedge\ pc[self] = \text{“UNINSTALL\_APP”} \\ \qquad\qquad\qquad\qquad\quad \wedge\ env\_vars' = [env\_vars \text{ EXCEPT } !.applications[app\_u[self]].installed = \text{FALSE}] \\ \qquad\qquad\qquad\qquad\quad \wedge\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc] \\ \qquad\qquad\qquad\qquad\quad \wedge\ app\_u' = [app\_u \text{ EXCEPT } ![self] = Head(stack[self]).app\_u] \\ \qquad\qquad\qquad\qquad\quad \wedge\ stack' \ = [stack \text{ EXCEPT } ![self] \ = Tail(stack[self])] \\ \qquad\qquad\qquad\qquad\quad \wedge \text{ UNCHANGED } \langle app\_vars,\ aps\_vars,\ permission\_type, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad app\_,\ app\_up,\ app\_t,\ app\_d,\ perm\_, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad app\_r,\ app,\ perm,\ applications,\ i\_,\ i\rangle \end{array}$$

$$uninstallApp(self) \triangleq UNINSTALL\_APP(self)$$

$$\begin{array}{l} UPDATE\_APP(self) \triangleq \ \wedge\ pc[self] = \text{“UPDATE\_APP”} \\ \qquad\qquad\qquad\qquad\quad \wedge\ env\_vars' = [env\_vars \text{ EXCEPT } !.applications[app\_up[self]].version = env\_vars.ap \\ \qquad\qquad\qquad\qquad\quad \wedge\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc] \\ \qquad\qquad\qquad\qquad\quad \wedge\ app\_up' = [app\_up \text{ EXCEPT } ![self] = Head(stack[self]).app\_up] \\ \qquad\qquad\qquad\qquad\quad \wedge\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])] \\ \qquad\qquad\qquad\qquad\quad \wedge \text{ UNCHANGED } \langle app\_vars,\ aps\_vars,\ permission\_type,\ app\_, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad app\_u,\ app\_t,\ app\_d,\ perm\_,\ app\_r,\ app, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad perm,\ applications,\ i\_,\ i\rangle \end{array}$$

$$updateApp(self) \triangleq UPDATE\_APP(self)$$

$$\begin{array}{l} TERMINATED(self) \triangleq \ \wedge\ pc[self] = \text{“TERMINATED”} \\ \qquad\qquad\qquad\qquad\quad \wedge\ env\_vars' = [env\_vars \text{ EXCEPT } !.applications[self].terminated = \text{TRUE}] \\ \qquad\qquad\qquad\qquad\quad \wedge\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc] \\ \qquad\qquad\qquad\qquad\quad \wedge\ app\_t' = [app\_t \text{ EXCEPT } ![self] = Head(stack[self]).app\_t] \\ \qquad\qquad\qquad\qquad\quad \wedge\ stack' \ = [stack \text{ EXCEPT } ![self] \ = Tail(stack[self])] \\ \qquad\qquad\qquad\qquad\quad \wedge \text{ UNCHANGED } \langle app\_vars,\ aps\_vars,\ permission\_type,\ app\_, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad app\_u,\ app\_up,\ app\_d,\ perm\_,\ app\_r,\ app, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad perm,\ applications,\ i\_,\ i\rangle \end{array}$$

$$terminate(self) \triangleq TERMINATED(self)$$

$DECLARE\_PERMISSION(self) \triangleq$
$\land\ pc[self] = \text{“DECLARE\_PERMISSION”}$
$\land\ app\_vars' = [app\_vars \text{ EXCEPT } ![app\_d[self]].manifest[perm\_[self]] =$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app\_d' = [app\_d \text{ EXCEPT } ![self] = Head(stack[self]).app\_d]$
$\land\ perm\_' = [perm\_ \text{ EXCEPT } ![self] = Head(stack[self]).perm\_]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land \text{ UNCHANGED } \langle env\_vars, aps\_vars,$
$permission\_type, app\_, app\_u,$
$app\_up, app\_t, app\_r, app, perm,$
$applications, i\_, i\rangle$

$declarePermission(self) \triangleq DECLARE\_PERMISSION(self)$

$REVOKE\_PERMISSION(self) \triangleq$
$\land\ pc[self] = \text{“REVOKE\_PERMISSION”}$
$\land\ aps\_vars' = [aps\_vars \text{ EXCEPT } !.app\_permissions[app\_r[self]] = [p \in P$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app\_r' = [app\_r \text{ EXCEPT } ![self] = Head(stack[self]).app\_r]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land \text{ UNCHANGED } \langle env\_vars, app\_vars, permission\_type,$
$app\_, app\_u, app\_up, app\_t, app\_d,$
$perm\_, app, perm, applications, i\_,$
$i\rangle$

$revokePermission(self) \triangleq REVOKE\_PERMISSION(self)$

$REQUEST\_PERMISSION(self) \triangleq$
$\land\ pc[self] = \text{“REQUEST\_PERMISSION”}$
$\land \text{ IF } aps\_vars.app\_permissions[app[self]][perm[self]] \neq Null$
$\text{THEN } \land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app' = [app \text{ EXCEPT } ![self] = Head(stack[self]).app]$
$\land\ perm' = [perm \text{ EXCEPT } ![self] = Head(stack[self]).perm]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“SWITCH\_PERMISSION”}]$
$\land \text{ UNCHANGED } \langle stack, app, perm\rangle$
$\land \text{ UNCHANGED } \langle env\_vars, app\_vars, aps\_vars,$
$permission\_type, app\_, app\_u,$
$app\_up, app\_t, app\_d, perm\_, app\_r,$
$applications, i\_, i\rangle$

$SWITCH\_PERMISSION(self) \triangleq$
$\land\ pc[self] = \text{“SWITCH\_PERMISSION”}$
$\land\ \exists\, x \in PermissionType :$
$\land\ permission\_type' = x$
$\land \text{ IF } permission\_type' = PERMISSION\_TYPE\_NORMAL \lor permissi$
$\lor\ permission\_type' = PERMISSION\_TYPE\_SPECIAL$
$\text{THEN } \land \text{ TRUE}$
$\land \text{ UNCHANGED } aps\_vars$
$\text{ELSE } \land \text{ IF } \quad permission\_type' = PERMISSION\_TYPE\_RU$

$\lor\ permission\_type' = PERMISSION\_TYPE\_CUST$
THEN $\land \exists\, r \in \{\text{TRUE, FALSE}\}:$
$aps\_vars' = [aps\_vars \text{ EXCEPT } !.app\_pe$
ELSE $\land$ TRUE
$\land$ UNCHANGED aps_vars
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ app' = [app \text{ EXCEPT } ![self] = Head(stack[self]).app]$
$\land\ perm' = [perm \text{ EXCEPT } ![self] = Head(stack[self]).perm]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land$ UNCHANGED $\langle env\_vars,\ app\_vars,\ app\_,\ app\_u,$
$app\_up,\ app\_t,\ app\_d,\ perm\_,\ app\_r,$
$applications,\ i\_,\ i\rangle$

$requestPermission(self) \triangleq REQUEST\_PERMISSION(self)$
$\lor\ SWITCH\_PERMISSION(self)$

$EnvBegin \triangleq \land\ pc[EnvironmentId] = \text{“EnvBegin”}$
$\land\ \lor\ \land$ TRUE
$\land\ pc' = [pc \text{ EXCEPT } ![EnvironmentId] = \text{“EnvBegin”}]$
$\land$ UNCHANGED $\langle stack,\ app\_r\rangle$
$\lor\ \land \exists\, a \in Apps:$
$\land\ \land\ app\_r' = [app\_r \text{ EXCEPT } ![EnvironmentId] = a]$
$\land\ stack' = [stack \text{ EXCEPT } ![EnvironmentId] = \langle[procedure \mapsto \text{“revokePermission”},$
$pc \mapsto \text{“EnvBegin”},$
$app\_r \mapsto app\_r[Environment$
$\circ\ stack[EnvironmentId]]$
$\land\ pc' = [pc \text{ EXCEPT } ![EnvironmentId] = \text{“REVOKE\_PERMISSION”}]$
$\land$ UNCHANGED $\langle env\_vars,\ app\_vars,\ aps\_vars,\ permission\_type,$
$app\_,\ app\_u,\ app\_up,\ app\_t,\ app\_d,\ perm\_,\ app,$
$perm,\ applications,\ i\_,\ i\rangle$

$EnvNext \triangleq EnvBegin$

$AppBegin(self) \triangleq \land\ pc[self] = \text{“AppBegin”}$
$\land$ IF $env\_vars.applications[self].terminated \neq$ TRUE
THEN $\land\ \lor\ \land\ \land\ app\_t' = [app\_t \text{ EXCEPT } ![self] = self]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle[procedure \mapsto \text{“terminate”},$
$pc \mapsto \text{“AppBegin”},$
$app\_t \mapsto app\_t[self]]\rangle$
$\circ\ stack[self]]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“TERMINATED”}]$
$\land$ UNCHANGED $\langle app\_u,\ app\_up\rangle$
$\lor\ \land$ IF $env\_vars.applications[self].installed =$ TRUE
THEN $\land\ \lor\ \land\ \land\ app\_u' = [app\_u \text{ EXCEPT } ![self] = self]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle[procedure \mapsto$
$pc \mapsto$

$app\_u \quad \mapsto$
$\circ\ stack[self]]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“UNINSTALL\_APP”}]$
$\land \text{ UNCHANGED } app\_up$
$\lor\ \land \text{ IF } env\_vars.applications[self].version < MaxAllowed$
$\text{THEN } \land\ \land\ app\_up' = [app\_up \text{ EXCEPT } ![self] =$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [pr$
pc
ap
$\circ\ s$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“UPDATE\_A}$
$\text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“AppBegin”}]$
$\land \text{ UNCHANGED } \langle stack,$
$app\_up \rangle$
$\land\ app\_u' = app\_u$
$\text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“DECLARING\_PERMISSION”}]$
$\land \text{ UNCHANGED } \langle stack,\ app\_u,$
$app\_up \rangle$
$\land\ app\_t' = app\_t$
$\text{ELSE } \land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“Done”}]$
$\land \text{ UNCHANGED } \langle stack,\ app\_u,\ app\_up,\ app\_t \rangle$
$\land \text{ UNCHANGED } \langle env\_vars,\ app\_vars,\ aps\_vars,$
$permission\_type,\ app\_,\ app\_d,\ perm\_,\ app\_r,$
$app,\ perm,\ applications,\ i\_,\ i \rangle$

$DECLARING\_PERMISSION(self) \triangleq\ \land\ pc[self] = \text{“DECLARING\_PERMISSION”}$
$\land\ \exists\, p \in Permissions :$
$\land\ \land\ app\_d' = [app\_d \text{ EXCEPT } ![self] = self]$
$\land\ perm\_' = [perm\_ \text{ EXCEPT } ![self] = p]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto \text{“declareP}$
$pc \mapsto \text{“INSTALL}$
$app\_d \mapsto app\_d[se$
$perm\_ \mapsto perm\_[sel$
$\circ\ stack[self]]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“DECLARE\_PERMISSION”}]$
$\land \text{ UNCHANGED } \langle env\_vars,\ app\_vars,\ aps\_vars,$
$permission\_type,\ app\_,\ app\_u,$
$app\_up,\ app\_t,\ app\_r,\ app,\ perm,$
$applications,\ i\_,\ i \rangle$

$INSTALLING\_APP(self) \triangleq\ \land\ pc[self] = \text{“INSTALLING\_APP”}$
$\land\ \land\ app\_' = [app\_ \text{ EXCEPT } ![self] = self]$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto \text{“installApp”},$
$pc \mapsto \text{“AppBegin”},$
$app\_ \mapsto app\_[self]] \rangle$

$$\circ\ stack[self]]$$
$$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“INSTALL\_APP”}]$$
$$\land \text{ UNCHANGED } \langle env\_vars,\ app\_vars,\ aps\_vars,$$
$$permission\_type,\ app\_u,\ app\_up,\ app\_t,$$
$$app\_d,\ perm\_,\ app\_r,\ app,\ perm,$$
$$applications,\ i\_,\ i\rangle$$

$$AppNext(self) \triangleq AppBegin(self) \lor DECLARING\_PERMISSION(self)$$
$$\lor INSTALLING\_APP(self)$$

$$ApsBegin \triangleq \land pc[ApsId] = \text{“ApsBegin”}$$
$$\land \lor \land \text{TRUE}$$
$$\land pc' = [pc \text{ EXCEPT } ![ApsId] = \text{“ApsBegin”}]$$
$$\land \text{UNCHANGED } \langle stack,\ app,\ perm\rangle$$
$$\lor \land \exists\, a \in Apps :$$
$$\exists\, p \in Permissions :$$
$$\land \land app' = [app \text{ EXCEPT } ![ApsId] = a]$$
$$\land perm' = [perm \text{ EXCEPT } ![ApsId] = p]$$
$$\land stack' = [stack \text{ EXCEPT } ![ApsId] = \langle [procedure \mapsto \text{“requestPermission”},$$
$$pc \mapsto \text{“ApsBegin”},$$
$$app \mapsto app[ApsId],$$
$$perm \mapsto perm[ApsId]]\rangle$$
$$\circ\ stack[ApsId]]$$
$$\land pc' = [pc \text{ EXCEPT } ![ApsId] = \text{“REQUEST\_PERMISSION”}]$$
$$\land \text{UNCHANGED } \langle env\_vars,\ app\_vars,\ aps\_vars,\ permission\_type,$$
$$app\_,\ app\_u,\ app\_up,\ app\_t,\ app\_d,\ perm\_,\ app\_r,$$
$$applications,\ i\_,\ i\rangle$$

$$ApsNext \triangleq ApsBegin$$

$$Next \triangleq EnvNext \lor ApsNext$$
$$\lor (\exists\, self \in ProcSet : \lor installApp(self) \lor uninstallApp(self)$$
$$\lor updateApp(self) \lor terminate(self)$$
$$\lor declarePermission(self)$$
$$\lor revokePermission(self)$$
$$\lor requestPermission(self))$$
$$\lor (\exists\, self \in Apps : AppNext(self))$$

$$Spec \triangleq \land Init \land \Box[Next]_{vars}$$
$$\land \land \text{WF}_{vars}((pc[EnvironmentId] \neq \text{“EnvBegin”}) \land EnvNext)$$
$$\land \text{WF}_{vars}(revokePermission(EnvironmentId))$$
$$\land \forall\, self \in Apps : \land \text{WF}_{vars}((pc[self] \neq \text{“AppBegin”}) \land AppNext(self))$$
$$\land \text{WF}_{vars}(terminate(self))$$
$$\land \text{WF}_{vars}(uninstallApp(self))$$
$$\land \text{WF}_{vars}(updateApp(self))$$
$$\land \text{WF}_{vars}(declarePermission(self))$$

$$\begin{array}{l} \qquad\qquad \wedge \mathrm{WF}_{vars}(installApp(self)) \\ \wedge \ \wedge \mathrm{WF}_{vars}((pc[ApsId] \neq \text{“ApsBegin”}) \wedge ApsNext) \\ \qquad \wedge \mathrm{WF}_{vars}(requestPermission(ApsId)) \end{array}$$

END TRANSLATION

* Modification History
* Last modified Sun May 21 05:35:10 $GMT + 03{:}30$ 2023 by *Amirhosein*
* Created *Fri Apr* 28 08:40:56 $GMT + 03{:}30$ 2023 by *Amirhosein*