

**MACHINE** ATM\_R1

**REFINES** ATM\_A

**SEES** ATM\_R1\_implicitContext

**VARIABLES**

**account**    refined class instances  
**atm**        class instances  
**bal**        inherited attribute of account  
**atm\_acbal**    attribute of atm  
**atm\_cash**    attribute of atm  
**atm\_card**    attribute of atm  
**idle**        state from statemachine, ATM.SM  
**active\_atm**    state from statemachine, ATM.SM

**INVARIANTS**

**atm.type** :  $atm \in \mathbb{P}(ATM)$   
**atm\_acbal.type** :  $atm\_acbal \in atm \leftrightarrow \mathbb{N}$   
**atm\_cash.type** :  $atm\_cash \in atm \rightarrow \mathbb{N}$   
**atm\_card.type** :  $atm\_card \in atm \leftrightarrow ValidCard$   
**idle.type** :  $idle \in \mathbb{P}(atm)$   
**active\_atm.type** :  $active\_atm \in \mathbb{P}(atm)$   
**ATM.SM.partitions\_atm** :  $partition(atm, idle, active\_atm)$

**EVENTS**

**Initialisation**

**begin**  
  **account.init** :  $account := \emptyset$   
  **atm.init** :  $atm := \emptyset$   
  **bal.init** :  $bal := \emptyset$   
  **atm\_acbal.init** :  $atm\_acbal := \emptyset$   
  **atm\_cash.init** :  $atm\_cash := \emptyset$   
  **atm\_card.init** :  $atm\_card := \emptyset$   
  **idle.init** :  $idle := \emptyset$   
  **active\_atm.init** :  $active\_atm := \emptyset$   
**end**

**Event** *start*  $\hat{=}$

**any**  
  *selfATM*    constructed instance of class atm  
**where**  
  **selfATM.type** :  $selfATM \in ATM \setminus atm$   
**then**  
  **atm\_constructor** :  $atm := atm \cup \{selfATM\}$

```

    atm.atm_cash_initialise : atm_cash(selfATM) := MAX_CASH
    ATM_SM_enterState_idle : idle := idle  $\cup$  {selfATM}
end

Event insertCard  $\hat{=}$ 

    any
        selfATM      contextual instance of class atm
        c
    where
        selfATM.type : selfATM  $\in$  atm
        c.type : c  $\in$  ValidCard
        ATM_SM_isin_idle : selfATM  $\in$  idle
        insertCard.Guard1 : selfATM  $\notin$  dom(atm_card)
    then
        ATM_SM_leaveState_idle : idle := idle  $\setminus$  {selfATM}
        ATM_SM_enterState_active_atm : active_atm := active_atm  $\cup$  {selfATM}
        insertCard.Action1 : atm_card := atm_card  $\cup$  {selfATM  $\mapsto$  c}
    end

Event reloadCash  $\hat{=}$ 

    any
        selfATM      contextual instance of class atm
    where
        selfATM.type : selfATM  $\in$  atm
        ATM_SM_isin_idle : selfATM  $\in$  idle
        reloadCash.Guard1 : atm_cash(selfATM) < MIN_CASH
    then
        reloadCash.Action1 : atm_cash(selfATM) := MAX_CASH - atm_cash(selfATM)
    end

Event withdrawOK  $\hat{=}$ 

refines withdraw

    any
        selfATM      contextual instance of class atm
        c
        am
        ac
    where
        am.type : am  $\in$   $\mathbb{N}$ 
        ac.type : ac  $\in$  account
        selfATM.type : selfATM  $\in$  atm
        c.type : c  $\in$  ValidCard
        ATM_SM_isin_active_atm : selfATM  $\in$  active_atm
        withdrawOK.Guard1 : selfATM  $\in$  dom(atm_card)
        withdrawOK.Guard2 : atm_card(selfATM) = c
        withdrawOK.Guard3 : bal(ac)  $\geq$  am
        withdrawOK.Guard4 : card_account(c) = ac
        withdrawOK.Guard5 : atm_cash(selfATM)  $\geq$  am

```

```

with
  self : ac = self
then
  withdrawOK.Action1 :  $bal(ac) := bal(ac) - am$ 
  withdrawOK.Action2 :  $atm\_acbal(selfATM) := bal(ac)$ 
  withdrawOK.Action3 :  $atm\_cash(selfATM) := atm\_cash(selfATM) - am$ 
end

Event withdrawFail  $\hat{=}$ 

  any
    selfATM    contextual instance of class atm
    c
    am
    ac
  where
    am.type :  $am \in \mathbb{N}$ 
    ac.type :  $ac \in account$ 
    selfATM.type :  $selfATM \in atm$ 
    c.type :  $c \in ValidCard$ 
    ATM.SM.isin_active_atm :  $selfATM \in active\_atm$ 
    withdrawFail.Guard1 :  $selfATM \in dom(atm\_card)$ 
    withdrawFail.Guard2 :  $atm\_card(selfATM) = c$ 
    withdrawFail.Guard3 :  $card\_account(c) = ac$ 
    withdrawFail.Guard4 :  $bal(ac) < am$ 
  then
    withdrawFail.Action1 :  $atm\_acbal(selfATM) := bal(ac)$ 
  end

Event checkBalance  $\hat{=}$ 

refines checkBalance

  any
    selfATM    contextual instance of class atm
    c
    ac
  where
    selfATM.type :  $selfATM \in atm$ 
    c.type :  $c \in ValidCard$ 
    ac.type :  $ac \in account$ 
    ATM.SM.isin_active_atm :  $selfATM \in active\_atm$ 
    checkBalance.Guard1 :  $selfATM \in dom(atm\_card)$ 
    checkBalance.Guard2 :  $atm\_card(selfATM) = c$ 
    checkBalance.Guard3 :  $card\_account(c) = ac$ 
  with
    self : ac = self
  then
    checkBalance.Action1 :  $atm\_acbal(selfATM) := bal(ac)$ 
  end

```

**Event** *ejectCard*  $\hat{=}$

**any**

*selfATM*      contextual instance of class atm  
*c*

**where**

*c.type* : *c*  $\in$  *ValidCard*  
*selfATM.type* : *selfATM*  $\in$  *atm*  
*ATM.SM.isin\_active\_atm* : *selfATM*  $\in$  *active\_atm*  
*ejectCard.Guard1* : *selfATM*  $\in$  *dom(atm\_card)*  
*ejectCard.Guard2* : *atm\_card*(*selfATM*) = *c*

**then**

*ATM.SM.leaveState\_active\_atm* : *active\_atm* := *active\_atm*  $\setminus$  {*selfATM*}  
*ATM.SM.enterState\_idle* : *idle* := *idle*  $\cup$  {*selfATM*}  
*ejectCard.Action1* : *atm\_card* := *atm\_card*  $\setminus$  {*selfATM*  $\mapsto$  *c*}

**end**

**Event** *createAccount*  $\hat{=}$

**refines** *createAccount*

**any**

*self*      contextual instance of refined class account

**where**

*self.type* : *self*  $\in$  *account*

**then**

*skip*

**end**

**Event** *deposit*  $\hat{=}$

**refines** *deposit*

**any**

*self*      contextual instance of refined class account

**where**

*self.type* : *self*  $\in$  *account*

**then**

*skip*

**end**

**END**