

Bancaire - V1 (+prm)

Documentation de projet

C. Attiogbe

20 Avril 2004

1 INTRODUCTION

La documentation est effectuée sur le projet : New1Bancaire.

2 STRUCTURE GENERALE

2.1 Introduction

2.2 Graphe de dépendance

File New1Bancaire.ps doesn't exist.

2.3 Métrique

PROJECT STATUS

COMPONENT	TC	POG	Obv	nPO	nUn	%Pr	B0C	C	Ada	C++	HIA
APPLICATION	OK	OK	3	0	0	100	-				
APPLICATION_I	OK	OK	1558	302	0	100	OK	OK	-	-	-
BANQUE	OK	OK	62	37	0	100	-				
BANQUE_ETENDUE	OK	OK	23	0	0	100	-				
BANQUE_ETENDUE_I	OK	OK	354	70	27	61	-	-	-	-	-
BANQUE_I	OK	OK	459	338	49	85	-	-	-	-	-
INTERFACE_BANQUE	OK	OK	11	0	0	100	-				
INTERFACE_BANQUE_I	OK	OK	130	81	47	41	-	-	-	-	-
PrmBANQUE	OK	OK	5	0	0	100	-				
PrmBANQUE_I	OK	OK	22	2	0	100	-	-	-	-	-
REQUETES	OK	OK	11	0	0	100	-				
REQUETES_I	OK	OK	41	20	0	100	-	-	-	-	-
UTILITAIRES	OK	OK	9	0	0	100	-				
UTILITAIRES_I	OK	OK	16	12	0	100	-	-	-	-	-
TOTAL	OK	OK	2704	862	123	85	-	-	-	-	-

3 DESCRIPTION DES COMPOSANTS

3.1 Composant APPLICATION

3.1.1 Présentation

Le composant traité dans ce paragraphe est : APPLICATION.

3.1.2 Métrique

APPLICATION AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/APPLICA

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
Initialisation	1	0	0	0	0	100
main	2	0	0	0	0	100
APPLICATION	3	0	0	0	0	100

3.1.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/APPLICATION.pmm non existant.

3.1.4 Source B

MACHINE

APPLICATION

OPERATIONS

main =

BEGIN

skip

END

END

3.2 Composant APPLICATION_I

3.2.1 Présentation

Le composant traité dans ce paragraphe est : APPLICATION_I.

3.2.2 Métrique

APPLICATION_I AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/APPLIC

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	1	0	0	0	0	100
InstanciatedConstraintsLemmas	3	0	0	0	0	100
Initialisation	1	0	0	0	0	100
main	1553	302	0	302	0	100
APPLICATION_I	1558	302	0	302	0	100

3.2.3 Règles ajoutées à la preuve

Fichier */home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/APPLICATION_I.pmm*
non existant.

3.2.4 Source B

IMPLEMENTATION

APPLICATION_I

REFINES

APPLICATION

IMPORTS

INTERFACE_BANQUE(10100)

, *PrmBANQUE*

, *BASIC_IO*

OPERATIONS

main =

VAR *rr, xx, res* **IN**

xx := **MAXINT**

; *rr* := 1

; *res* := *echec*

; **WHILE** (*rr* ≠ 0) ∧ (*xx* ≠ 0)

DO

STRING_WRITE(" \t + _____ + \n")

; **STRING_WRITE**(" \t + Menu de l'application Bancaire + \n")

; **STRING_WRITE**(" \t + _____ + \n")

; **STRING_WRITE**(" \t Nouveau client : 1 \n")

; **STRING_WRITE**(" \t Nouveau compte : 2 \n")

; **STRING_WRITE**(" \t Interrogation d'un compte : 3 \n")

; **STRING_WRITE**(" \t Approvisionnement d'un compte : 4 \n")

; **STRING_WRITE**(" \t Retrait d'un compte : 5 \n")

; **STRING_WRITE**(" \t Transfert entre deux comptes : 6 \n")

; **STRING_WRITE**(" \t Fermeture d'un compte : 7 \n")

; **STRING_WRITE**(" \t Suppression d'un client : 8 \n")

; **STRING_WRITE**(" \t + _____ + \n")

; **STRING_WRITE**(" \t Quitter : 0 \n")

; **STRING_WRITE**(" \t + _____ + \n \n")

; **STRING_WRITE**(" \t choix ? ")

; *rr* ← **INTERVAL_READ**(0,8)

; **CASE** *rr* **OF**

EITHER 1 **THEN**

res ← *nouveau_client*

OR 2 **THEN**

res ← *nouveau_compte*

OR 3 **THEN**

res ← *interrogation_compte*

OR 4 **THEN**

```

        res ← approvisionnement_compte
    OR 5 THEN

        res ← retrait_compte
    OR 6 THEN

        res ← transfert_comptes
    OR 7 THEN

        res ← fermeture_compte
    OR 8 THEN

        res ← suppression_client
    END
END
;

CASE res OF
    EITHER succes
        THEN STRING_WRITE("")
    OR echec
        THEN
            STRING_WRITE("Echec")
    OR plus_argent
        THEN
            STRING_WRITE("Il n'y a plus d'argent sur ce compte")
    OR client_non_valable
        THEN
            STRING_WRITE("Ce numero de client n'est pas attribue")
    OR compte_non_valable
        THEN
            STRING_WRITE("Ce numero de compte n'est pas attribue")
    OR plus_de_compte_disponible
        THEN
            STRING_WRITE("Il n'y a plus de numero de compte disponible")
    OR plus_de_client_disponible
        THEN
            STRING_WRITE("plus_de_client_disponible")
    OR plafond_atteint
        THEN
            STRING_WRITE("Vous allez depasser la limite d'argent")
    OR decouvert_depasse
        THEN
            STRING_WRITE("Vous etes a decouvert !")
    OR comptes_identiques
        THEN
            STRING_WRITE("Les deux comptes sont identiques !")
    OR compte_non_solde
        THEN
            STRING_WRITE("Compte non soldé (solde /= 0), fermeture impossible")
    ELSE
        STRING_WRITE("Résultat ininterprétable")

```

```

    END
  END
  ; STRING_WRITE(" \n")
  ; STRING_WRITE(" \n")

  ;  $xx := xx - 1$ 
  INVARIANT
     $xx \in \text{NAT}$ 
  VARIANT
     $xx$ 
  END
END
END

```

3.3 Composant BANQUE

3.3.1 Présentation

Le composant traité dans ce paragraphe est : BANQUE.

3.3.2 Métrique

BANQUE AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/BANQUE.mch

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
Initialisation	4	6	0	6	0	100
nouveaucient	3	9	0	9	0	100
approvisionnement	8	3	1	2	0	100
retrait	8	3	0	3	0	100
transfert	8	3	0	3	0	100
solde_compte	11	0	0	0	0	100
decouvert_compte	11	0	0	0	0	100
fermeturecompte	1	10	0	10	0	100
suppressionclient	8	3	0	3	0	100
BANQUE	62	37	1	36	0	100

3.3.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/BANQUE.pmm non existant.

3.3.4 Source B

MACHINE

BANQUE(idbanque)

CONSTRAINTS

idbanque $\in \text{NAT}$

\wedge *idbanque* > 10000

SEES

PrmBANQUE

CONCRETE_CONSTANTS

nbcompte
, *nbclient*
, *decspec*
, *soldpec*
, *comptspec*
, *clispec*

PROPERTIES

nbcompte ∈ **INT**
∧ *nbclient* ∈ **INT**
∧ *nbclient* < **MAXINT**
∧ *nbcompte* < **MAXINT**
∧ *decspec* ∈ **INT**
∧ *decspec* < **MAXINT**
∧ **MININT** < *decspec*
∧ *soldpec* ∈ **INT**
∧ *soldpec* < **MAXINT**
∧ *comptspec* ∈ *NOCOMPTE*
∧ *clispec* ∈ *NOCLIENT*

VARIABLES

porteur
, *decouvert*
, *solde*
, *lescomptes*

INVARIANT

lescomptes ⊆ *NOCOMPTE*
∧ *porteur* ∈ *lescomptes* → *NOCLIENT*
∧ *decouvert* ∈ *lescomptes* → **INT**
∧ *solde* ∈ *lescomptes* → **INT**

∧ { *comptspec* ↦ *clispec* } ⊆ *porteur*
∧ { *comptspec* ↦ *decspec* } ⊆ *decouvert*
∧ { *comptspec* ↦ *soldpec* } ⊆ *solde*

INITIALISATION

ANY *soldp*, *decsp* **WHERE**

soldp ∈ **INT**
∧ *decsp* ∈ **INT**

THEN

porteur := { *comptspec* ↦ *clispec* }
|| *decouvert* := { *comptspec* ↦ *decspec* }
|| *solde* := { *comptspec* ↦ *soldpec* }


```

||   lescomptes := {comptspec}
END

```

OPERATIONS

```

nouveaucient(cli, com) =
    /* Enregistrer un cli+compte dans porteur */
    /* un decouvert par default */
    /*-----*/

```

PRE

```

    cli ∈ NOCLIANT
    ∧ cli ≠ clispec
    ∧ com ∈ NOCOMPTE
    ∧ com ∉ lescomptes
    ∧ com ≠ comptspec

```

THEN

```

    lescomptes := lescomptes ∪ {com}
    || porteur := porteur ∪ {com ↦ cli}
    || solde := solde ∪ {com ↦ 0}
    || decouvert := decouvert ∪ {com ↦ decspec}

```

END

```

;
approvisionnement (compt,som) =

```

```

    /*-----*/

```

PRE

```

    som ∈ INT
    ∧ som > 0 ∧ som < MAXINT
    ∧ compt ∈ NOCOMPTE
    ∧ compt ∈ lescomptes
    ∧ compt ≠ comptspec
    ∧ (solde(compt) + som) < MAXINT
    ∧ MININT < (solde(compt) + som)

```

THEN

```

solde(compt) := solde(compt) + som

```

END

```

;
retrait(compt,som) =

```

```

    /* retrait d'argent d'un compte */
    /* cas ou la somme à retirer n'amène pas */
    /* le compte à découvert */
    /*-----*/

```

PRE

```

    som ∈ INT ∧ som > 0 ∧ som < MAXINT
    ∧ compt ∈ NOCOMPTE
    ∧ compt ∈ lescomptes
    ∧ compt ≠ comptspec
    ∧ MININT ≤ (solde(compt) - som)
    ∧ (solde(compt) - som) ≤ MAXINT
    ∧ (solde(compt) - som) > decouvert(compt)

```

THEN

```

    solde := ({compt} ↦ solde) ∪ {compt ↦ (solde(compt) - som)}

```

```

END
;
transfert(compt1, compt2, som) =
/* transfert d'une somme d'argent d'un compte
vers un autre compt */
/*-----*/
PRE
  compt1 ∈ NOCOMPTE ∧ compt1 ∈ lescomptes
  ∧ compt2 ∈ NOCOMPTE ∧ compt2 ∈ lescomptes
  ∧ compt1 ≠ compt2
  ∧ compt1 ≠ comptspec
  ∧ compt2 ≠ comptspec
  ∧ som ∈ INT ∧ som > 0 ∧ som < MAXINT
  ∧ MININT < (solde(compt1) - som)
  ∧ (solde(compt1) - som) < MAXINT
  ∧ (solde(compt1) - som) > decouvert(compt1)
  ∧ (solde(compt2) + som) < MAXINT
  ∧ MININT < (solde(compt2) + som)
THEN

  solde := ({compt2} ◁ ({compt1} ◁ solde)) ∪
  {compt1 ↦ (solde(compt1) - som)} ∪
  {compt2 ↦ (solde(compt2) + som)}

END
;
rsom ← solde_compte (compt) =
/* donner le solde d'un compte indiqué */
/*-----*/
PRE
  compt ∈ NOCOMPTE ∧ compt ∈ lescomptes
THEN
  rsom := solde(compt)
END
;
rdec ← decouvert_compte(compt) =
/* montant du decouvert autorisé sur un compte */
/*-----*/
PRE
  compt ∈ NOCOMPTE ∧ compt ∈ lescomptes
THEN
  rdec := decouvert(compt)
END
;
fermeturecompte (compt) =
/* fermeture d'un compte donné, d'un client */
/* le client peut avoir plusieurs comptes */
/* le compte doit etre solde !!! */
/*-----*/
PRE
  compt ∈ NOCOMPTE ∧ compt ∈ lescomptes

```

```

     $\wedge$  compt  $\neq$  comptspec
 $\wedge$  porteur(compt)  $\neq$  clispec
 $\wedge$  solde(compt) = 0
 $\wedge$  {compt  $\mapsto$  solde(compt)}  $\neq$  {comptspec  $\mapsto$  decspec}
THEN
    porteur := {compt}  $\Leftarrow$  porteur
    || solde := {compt}  $\Leftarrow$  solde
    || decouvert := {compt}  $\Leftarrow$  decouvert
    || lescomptes := lescomptes - {compt}
END
;
suppressionclient (cli) =
    /* suppression d'un client,
    tous les comptes sont déjà soldés et supprimés */
    /*-----*/
PRE
    cli  $\in$  NOCLIENT  $\wedge$  cli  $\in$  ran(porteur)
     $\wedge$  cli  $\neq$  clispec
 $\wedge$   $\emptyset =$  porteur-1 [{cli}]
THEN
    porteur := porteur  $\triangleright$  {cli}
END

```

END

3.4 Composant BANQUE_ETENDUE

3.4.1 Présentation

Le composant traité dans ce paragraphe est : *BANQUE_ETENDUE*.

3.4.2 Métrique

BANQUE_ETENDUE AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/BAN

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
Instancié	4	0	0	0	0	100
Initialisation	1	0	0	0	0	100
unclient_libre	3	0	0	0	0	100
uncompte_libre	3	0	0	0	0	100
s_compte_valide	2	0	0	0	0	100
s_client_valide	2	0	0	0	0	100
nouveaucient	1	0	0	0	0	100
approvisionnement	1	0	0	0	0	100
retrait	1	0	0	0	0	100
transfert	1	0	0	0	0	100
solde_compte	1	0	0	0	0	100
decouvert_compte	1	0	0	0	0	100
fermeturecompte	1	0	0	0	0	100
suppressionclient	1	0	0	0	0	100
BANQUE_ETENDUE	23	0	0	0	0	100

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
--	-------	------	-------	-------	------	-----

3.4.3 Règles ajoutées à la preuve

Fichier */home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/BANQUE_ETENDUE.pmm* non existant.

3.4.4 Source B

MACHINE

BANQUE_ETENDUE(idbanque)

CONSTRAINTS

idbanque ∈ NAT
 ∧ *idbanque* > 10000

EXTENDS

BANQUE(idbanque)

SEES

PrmBANQUE

CONCRETE_CONSTANTS

e_nbcompte
 , *e_nbclient*

PROPERTIES

e_nbcompte ∈ INT
 ∧ *e_nbclient* ∈ INT

OPERATIONS

cl, tcl ← *unclient_libre* =

CHOICE

cl :∈ NOCLIENT - ran(*porteur*)
 || *tcl* := TRUE
OR
cl :∈ NOCLIENT
 || *tcl* := FALSE
END

;
cp, tcp ← *uncompte_libre* =

CHOICE

cp :∈ NOCOMPTE - *lescomptes*
 || *tcp* := TRUE
OR
cp :∈ NOCOMPTE
 || *tcp* := FALSE
END

```

;
tt ← s_compte_valide (cpt)=

```

```

PRE
  cpt ∈ NOCOMPTE
THEN
  tt :∈ BOOL
END

```

```

;
tt ← s_client_valide(cl)=

```

```

PRE
  cl ∈ NOCLIENT
THEN
  tt :∈ BOOL
END

```

```

END

```

3.5 Composant BANQUE_ETENDUE_I

3.5.1 Présentation

Le composant traité dans ce paragraphe est : BANQUE_ETENDUE_I.

3.5.2 Métrique

BANQUE_ETENDUE_I POGenerated /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	27	10	0	10	0	100
InstanciatedConstraintsLemmas	15	20	0	17	3	85
Initialisation	5	10	0	1	9	10
unclient_libre	43	9	0	6	3	66
uncompte_libre	43	9	0	8	1	88
s_compte_valide	31	0	0	0	0	100
s_client_valide	31	0	0	0	0	100
nouveaucient	15	4	0	0	4	0
approvisionnement	23	1	0	0	1	0
retrait	24	1	0	0	1	0
transfert	30	1	0	0	1	0
solde_compte	17	0	0	0	0	100
decouvert_compte	17	0	0	0	0	100
fermeturecompte	16	4	0	0	4	0
suppressionclient	17	1	0	1	0	100
BANQUE_ETENDUE_I	354	70	0	43	27	61

3.5.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/BANQUE_ETENDUE_I.pmm non existant.

3.5.4 Source B

IMPLEMENTATION

BANQUE_ETENDUE_I(idbanque)

REFINES

BANQUE_ETENDUE

EXTENDS

BANQUE(idbanque)

SEES

PrmBANQUE

VALUES

e_nbcompte = cma-cmi
; *e_nbclient = clima - climi*

IMPORTS

porteurs.L_ARRAY3(NOCLIENT, e_nbcompte)
, *decouverts.L_ARRAY3(INT, e_nbcompte)*
, *soldes.L_ARRAY3(INT, e_nbcompte)*
, *clients.L_SET(e_nbclient, NOCLIENT)*
, *clientsl.L_SET(e_nbclient, NOCLIENT)*
, *comptes.L_SET(e_nbcompte, NOCOMPTE)*
, *comptesl.L_SET(e_nbcompte, NOCOMPTE)*

INVARIANT

ran(porteur) = **ran**(porteurs.arr_vrb)
∧ **ran**(decouvert) = **ran**(decouverts.arr_vrb)
∧ **ran**(solde) = **ran**(soldes.arr_vrb)
∧ **lescomptes** = **ran**(comptes.set_vrb)

∧ **ran**(comptesl.set_vrb) = NOCOMPTE - **ran**(comptes.set_vrb)

∧ **ran**(clientsl.set_vrb) = NOCLIENT - **ran**(clients.set_vrb)

∧ **dom**(porteurs.arr_vrb) = **dom**(comptes.set_vrb)
∧ **dom**(soldes.arr_vrb) = **dom**(comptes.set_vrb)
∧ **dom**(decouverts.arr_vrb) = **dom**(comptes.set_vrb)

OPERATIONS

cl, tcl ← *unclient_libre* =
 VAR *tr* **IN**
 tr := **FALSE**
 ; *tr* ← **clientsl.IS_FULL_SET**
 ;
 IF *tr* = **TRUE**
 THEN
 cl ← **to_Client**(*climi*)

```

;   tcl := FALSE
ELSE
   cl ← clientsl.VAL_SET(1)
;   clientsl.RMV_SET(cl)
;   tcl := TRUE
END
END
;
cp,tcp ← uncompte_libre =
  VAR tr IN
  tr := FALSE
; tr ← comptesl.IS_FULL_SET
;
IF tr = TRUE
THEN
  cp ← to_Compte(cmi)
; tcp := FALSE
ELSE
  cp ← comptesl.VAL_SET(1)
; comptesl.RMV_SET(cp)
; tcp := TRUE
END
END
;
tt ← s_compte_valide (cpt)=
  BEGIN
  VAR ind IN
  tt,ind ← comptes.FIND_SET(cpt)
  END
  END
;
tt ← s_client_valide(cl) =
  VAR ind IN
  tt,ind ← clients.FIND_SET(cl)
  END

```

END

3.6 Composant BANQUE_I

3.6.1 Présentation

Le composant traité dans ce paragraphe est : BANQUE_I.

3.6.2 Métrique

BANQUE_I POGenerated /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/BANQUE_I.

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	10	15	0	15	0	100

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
InstanciéConstraintsLemmas	10	20	0	19	1	95
Initialisation	81	135	0	125	10	92
nouveaucient	58	73	0	57	16	78
approvisionnement	34	14	0	12	2	85
retrait	34	12	0	10	2	83
transfert	76	23	0	19	4	82
solde_compte	34	3	0	2	1	66
decouvert_compte	34	3	0	2	1	66
fermeturecompte	40	31	0	19	12	61
suppressionclient	48	9	0	9	0	100
BANQUE_I	459	338	0	289	49	85

3.6.3 Règles ajoutées à la preuve

Fichier `/home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/BANQUE_I.pmm` non existant.

3.6.4 Source B

IMPLEMENTATION

BANQUE_I(idbanque)

REFINES

BANQUE

SEES

BASIC_IO

, *PrmBANQUE*

VALUES

nbcompte = *cma* - *cmi*
; *nbclient* = *clima* - *climi*
; *decspec* = 0
; *soldpec* = 0
; *comptspec* = *cma*
; *clispec* = *clima*

IMPORTS

porteurs.L_ARRAY3(NOCLIENT, nbcompte)
, *decouverts.L_ARRAY3(INT, nbcompte)*
, *soldes.L_ARRAY3(INT, nbcompte)*
, *clients.L_SET(nbclient, NOCLIENT)*
, *clientsl.L_SET(nbclient, NOCLIENT)*
, *comptes.L_SET(nbcompte, NOCOMPTE)*
, *comptesl.L_SET(nbcompte, NOCOMPTE)*

INVARIANT

cma+1 ≤ **MAXINT**
∧ *clima*+1 ≤ **MAXINT**
∧ *lescomptes* = **ran**(*comptes.set_vrb*)
∧ **dom**(*porteur*) = **ran**(*comptes.set_vrb*)
∧ **ran**(*porteur*) = **ran**(*porteurs.arr_vrb*)
∧ **dom**(*decouvert*) = **ran**(*comptes.set_vrb*)
∧ **ran**(*decouvert*) = **ran**(*decouverts.arr_vrb*)

\wedge $\mathbf{dom}(solde) = \mathbf{ran}(comptes.set_vrb)$
 \wedge $\mathbf{ran}(solde) = \mathbf{ran}(soldes.arr_vrb)$

\wedge $\mathbf{ran}(comptesl.set_vrb) = NOCOMPTE - \mathbf{ran}(comptes.set_vrb)$

\wedge $\mathbf{ran}(clientsl.set_vrb) = NOCLIENT - \mathbf{ran}(clients.set_vrb)$

\wedge $\mathbf{dom}(porteurs.arr_vrb) = \mathbf{dom}(comptes.set_vrb)$
 \wedge $\mathbf{dom}(soldes.arr_vrb) = \mathbf{dom}(comptes.set_vrb)$
 \wedge $\mathbf{dom}(decouverts.arr_vrb) = \mathbf{dom}(comptes.set_vrb)$

INITIALISATION

```

VAR ii, kk, cma1 IN
  ii := cmi
;   kk  $\leftarrow$  to_Compte(ii)
;   cma1 := cma+1
;   WHILE (ii < cma1) DO
  kk  $\leftarrow$  to_Compte(ii)
;   IF (kk  $\neq$  comptspec)
  THEN
    comptesl.INS_SET(kk)
  END
;   ii := ii+1
  INVARIANT
    ii  $\in$  cmi .. cma1
 $\wedge$  kk  $\in$  NOCOMPTE
 $\wedge$  cma1  $\leq$  MAXINT
  VARIANT
    cma1-ii
  END
END
;

```

```

VAR jj, kk, clima1 IN
  jj := climi
;   kk  $\leftarrow$  to_Client(jj)
;   clima1 := clima+1
;   WHILE (jj < clima1) DO
  kk  $\leftarrow$  to_Client(jj)
;   IF (kk  $\neq$  clispec)
  THEN
    clientsl.INS_SET(kk)
  END
;   jj := jj+1
  INVARIANT
    jj  $\in$  climi .. clima1
 $\wedge$  kk  $\in$  NOCLIENT
 $\wedge$  clima1  $\leq$  MAXINT
  VARIANT

```

```

        clima1-jj
    END
END

;   comptes.INS_SET(comptspec)
;   clients.INS_SET(clispec)

;   porteurs.STR_ARRAY(nbcompte,clispec)
;   decouverts.STR_ARRAY(nbcompte,decspec)
;   soldes.STR_ARRAY(nbcompte,soldpec)

```

OPERATIONS

nouveauclient (*cli, com*) =

```

    BEGIN
    comptes.INS_SET(com)
;   VAR tt, pos IN
        tt, pos ← comptes.FIND_SET(com)
;   IF tt = TRUE
    THEN
        porteurs.STR_ARRAY(pos, cli)
;       soldes.STR_ARRAY(pos, soldpec)
;       decouverts.STR_ARRAY(pos, decspec)
    END
    END
;   clients.INS_SET(cli)
END

```

;

approvisionnement (*compt,som*) =

```

    BEGIN
    VAR
        ancsolde, nouvsolde, tt, pos
    IN
        tt, pos ← comptes.FIND_SET(compt)
;       nouvsolde ← soldes.VAL_ARRAY(pos)
;       ancsolde ← soldes.VAL_ARRAY(pos)
;       IF tt = TRUE
    THEN
        ancsolde ← soldes.VAL_ARRAY(pos)
;       nouvsolde := ancsolde + som
;       soldes.STR_ARRAY(pos,nouvsolde)
    END
    END
    END

```

;

retrait (*compt,som*) =

```

BEGIN
VAR
    ancsolde, nouvsolde, tt, pos
IN
    tt, pos ← comptes.FIND_SET(compt)
;   IF tt = TRUE
    THEN
        ancsolde ← soldes.VAL_ARRAY(pos)
;       nouvsolde := ancsolde - som
;       soldes.STR_ARRAY(pos, nouvsolde)
    END
END
END
;
transfert(compt1, compt2, som) =
BEGIN
VAR
    ancsolde1, nouvsolde1, ancsolde2, nouvsolde2, tt1, tt2, pos1, pos2
IN
    tt1, pos1 ← comptes.FIND_SET(compt1)
;   tt2, pos2 ← comptes.FIND_SET(compt2)
;   IF tt1 = TRUE ∧ tt2 = TRUE
    THEN
        ancsolde1 ← soldes.VAL_ARRAY(pos1)
;       ancsolde2 ← soldes.VAL_ARRAY(pos2)
;       nouvsolde1 := ancsolde1 - som
;       nouvsolde2 := ancsolde2 + som
;       soldes.STR_ARRAY(pos1, nouvsolde1)
;       soldes.STR_ARRAY(pos2, nouvsolde2)
    END
END
END
END
;
rsom ← solde_compte (compt) =
BEGIN
VAR tt, pos IN
    tt, pos ← comptes.FIND_SET(compt)
;   IF tt = TRUE
    THEN
        rsom ← soldes.VAL_ARRAY(pos)
    END
END
END
;
rdec ← decouvert_compte(compt) =
BEGIN
VAR tt, pos IN
    tt, pos ← comptes.FIND_SET(compt)
;   IF tt = TRUE
    THEN

```

```

                rdec ← decouverts.VAL_ARRAY(pos)
            END
        END
    END
;
fermeturecompte (compt) =

    BEGIN
    VAR tt, pos IN
        tt, pos ← comptes.FIND_SET(compt)
    ; IF tt = TRUE
        THEN
            porteurs.STR_ARRAY(pos, clispec)
        ; soldes.STR_ARRAY(pos, soldpec)
        ; decouverts.STR_ARRAY(pos, decspec)
        ; comptes.RMV_SET(compt)
        ; comptesl.INS_SET(compt)
        END
    END
    END
;
suppressionclient (cli) =

    BEGIN
    VAR tt, pos IN
        tt, pos ← clients.FIND_SET(cli)
    ; IF tt = TRUE
        THEN
            clients.RMV_SET(cli)
        ; clientsl.INS_SET(cli)
        END
    END
    END
END

```

3.7 Composant INTERFACE_BANQUE

3.7.1 Présentation

Le composant traité dans ce paragraphe est : *INTERFACE_BANQUE*.

3.7.2 Métrique

INTERFACE_BANQUE AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/IN

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
Initialisation	1	0	0	0	0	100
code_en_message	2	0	0	0	0	100

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
nouveau_client	1	0	0	0	0	100
nouveau_compte	1	0	0	0	0	100
approvisionnement_compte	1	0	0	0	0	100
retrait_compte	1	0	0	0	0	100
transfert_comptes	1	0	0	0	0	100
interrogation_compte	1	0	0	0	0	100
fermeture_compte	1	0	0	0	0	100
suppression_client	1	0	0	0	0	100
INTERFACE_BANQUE	11	0	0	0	0	100

3.7.3 Règles ajoutées à la preuve

Fichier */home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/INTERFACE_BANQUE.pmm* non existant.

3.7.4 Source B

MACHINE

INTERFACE_BANQUE(idbanque)

CONSTRAINTS

$idbanque \in \mathbf{NAT} \wedge idbanque > 10000$

SETS

$REPONSE = \{succes, echec, plus_argent, client_non_valable, compte_non_valable, plus_de_compte_disponible, plus_de_client_disponible, plafond_atteint, decouvert_depasse, comptes_identiques, compte_non_solde\}$

OPERATIONS

$str \leftarrow code_en_message(res) =$

PRE

$res \in REPONSE$

THEN

$str : \in REPONSE$

END

;

$res \leftarrow nouveau_client =$

CHOICE

$res := succes$

OR

$res := plus_de_client_disponible$

OR

$res := plus_de_compte_disponible$

END

;

$res \leftarrow nouveau_compte =$

CHOICE

$res := succes$

OR

$res := plus_de_compte_disponible$

OR

$res := client_non_valable$

END

;

```

res ← approvisionnement_compte =
    CHOICE
        res := succes
    OR
        res := plafond_atteint
    OR
        res := compte_non_valable
    END
;
res ← retrait_compte =
    CHOICE
        res := succes
    OR
        res := decouvert_depasse
    OR
        res := compte_non_valable
    END
;
res ← transfert_comptes =
    CHOICE
        res := succes
    OR
        res := plafond_atteint
    OR
        res := decouvert_depasse
    OR
        res := comptes_identiques
    OR
        res := compte_non_valable
    END
;
res ← interrogation_compte =
    CHOICE
        res := succes
    OR
        res := compte_non_valable
    OR
        res := compte_non_solde
    END
;
res ← fermeture_compte =
    CHOICE
        res := succes
    OR
        res := compte_non_valable
    END
;
res ← suppression_client =
    CHOICE
        res := succes
    OR
        res := client_non_valable
    END

```

END

3.8 Composant INTERFACE_BANQUE_I

3.8.1 Présentation

Le composant traité dans ce paragraphe est : *INTERFACE_BANQUE_I*.

3.8.2 Métrique

INTERFACE_BANQUE_I POGenerated /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	2	0	0	0	0	100
InstanciatedConstraintsLemmas	5	0	0	0	0	100
Initialisation	2	0	0	0	0	100
code_en_message	4	0	0	0	0	100
nouveau_client	22	6	0	4	2	66
nouveau_compte	20	5	0	3	2	60
approvisionnement_compte	15	13	0	5	8	38
retrait_compte	15	14	0	6	8	42
transfert_comptes	24	25	0	9	16	36
interrogation_compte	6	6	0	4	2	66
fermeture_compte	9	8	0	2	6	25
suppression_client	6	4	0	1	3	25
INTERFACE_BANQUE_I	130	81	0	34	47	41

3.8.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/INTERFACE_BANQUE_I.pmm non existant.

3.8.4 Source B

IMPLEMENTATION

INTERFACE_BANQUE_I(idbanque)

REFINES

INTERFACE_BANQUE

IMPORTS

BANQUE_ETENDUE(idbanque)
, *REQUETES*

SEES

BASIC_IO

OPERATIONS

str ← *code_en_message*(*res*) =

BEGIN

str := *res*

END

;

```

res ← nouveau_client =
  VAR cli, bcli IN

    cli, bcli ← unclient_libre
  ; IF
    bcli = TRUE
  THEN
    VAR com, bcom IN

      com, bcom ← uncompte_libre
    ; IF
      bcom = TRUE
    THEN
      res := succes
      ; nouveauclient(cli, com)
      ; STRING_WRITE("Creation du nouveau client : ")
      ; affiche_client(cli)
      ; STRING_WRITE(" et de son compte : ")
      ; affiche_compte(com)
      ; STRING_WRITE(" effectuee \n")
      ; STRING_WRITE("\n")
    ELSE
      res := plus_de_compte_disponible
    END
  END
ELSE
  res := plus_de_client_disponible
END
END
;

res ← nouveau_compte =
  VAR cc, bb IN
    cc ← saisir_client
  ; bb ← s_client_valide(cc)
  ; IF
    bb = TRUE
  THEN
    VAR com IN
      com, bb ← uncompte_libre
    ; IF
      bb = TRUE
    THEN
      res := succes
      ; nouveauclient(cc, com)
      ; STRING_WRITE("Creation du nouveau compte : ")
      ; affiche_compte(com)
      ; STRING_WRITE(" effectuee pour le client : ")
      ; affiche_client(cc)
      ; STRING_WRITE("\n")
      ; STRING_WRITE("\n")
    ELSE

```



```

                                res := plus_de_compte_disponible
                                END
                                END
                                ELSE
                                res := client_non_valable
                                END
                                END
                                ;

res ← approvisionnement_compte =
    VAR cc, ss, bb IN
        cc ← saisir_compte
    ; ss ← saisir_somme
    ; bb ← s_compte_valide(cc)
    ; IF
        bb = TRUE
    THEN
        VAR scc, nsom IN
            scc ← solde_compte(cc)
        ; nsom := scc + ss
        ; IF
            nsom < MAXINT
        THEN
            res := succes
        ; approvisionnement (cc, ss)
        ; STRING_WRITE("Approvisionnement effectue \n")
        ELSE
            res := plafond_atteint
        END
        END
    ELSE
        res := compte_non_valable
    END
    END
    ;

res ← retrait_compte =
    VAR cc, ss, bb IN
        cc ← saisir_compte
    ; ss ← saisir_somme
    ; bb ← s_compte_valide(cc)
    ; IF
        bb = TRUE
    THEN
        VAR ass, dif, dec IN
            ass ← solde_compte(cc)
        ; dif := ass - ss
        ; dec ← decouvert_compte(cc)
        ; IF
            dif > dec
            ∧ dif > MININT
        THEN
            res := succes

```

```

; retrait(cc, ss)
; STRING_WRITE("Retrait effectue \n")
ELSE
    res := decouvert_depasse
END
END
ELSE
    res := compte_non_valable
END
END
;

res ← transfert_comptes =
VAR c1, c2, ss, bb1, bb2 IN
    c1 ← saisir_compte
; c2 ← saisir_compte
; ss ← saisir_somme
; bb1 ← s_compte_valide(c1)
; bb2 ← s_compte_valide(c2)
; IF
    bb1 = TRUE ∧ bb2 = TRUE
    THEN
        IF
            c1 ≠ c2
            THEN
                VAR sc1, dif, dec IN
                    sc1 ← solde_compte(c1)
; dif := sc1 - ss
; dec ← decouvert_compte(c1)
; IF
                    dif ≥ dec
                    ∧ dif ≥ MININT
                    THEN
                        VAR sc2, som IN
                            sc2 ← solde_compte(c2)
; som := sc2 + ss
; IF
                                som < MAXINT
                                THEN
                                    res := succes
; transfert(c1, c2, ss)
; STRING_WRITE("Transfert effectue\n")
                                ELSE
                                    res := plafond_atteint
                                END
                            END
                        END
                    ELSE
                        res := decouvert_depasse
                    END
                END
            ELSE
                res := comptes_identiques
            END
        END
    END

```

```

        ELSE
            res := compte_non_valable
        END
    END
END
;

res ← interrogation_compte =
    VAR cc, bb IN
        cc ← saisir_compte
    ; bb ← s_compte_valide(cc)
    ; IF
        bb = TRUE
    THEN
        VAR ss IN
            res := succes
            ; ss ← solde_compte(cc)
            ; STRING_WRITE("Le solde de votre compte est : ")
            ; INT_WRITE(ss)
            ; STRING_WRITE("\\n")
        END
    ELSE
        res := compte_non_valable
    END
END
;

res ← fermeture_compte =
    VAR cc, bb, ccsolde IN
        cc ← saisir_compte
    ; bb ← s_compte_valide(cc)
    ; ccsolde := 0
    ; IF
        bb = TRUE
    THEN
        ccsolde ← solde_compte(cc)
    ; IF ccsolde = 0
    THEN
        res := succes
        ; fermeturecompte(cc)
        ; STRING_WRITE("Fermeture du compte effectuee \\n")
    ELSE
        res := compte_non_solde
    ; STRING_WRITE("Fermeture impossible, compte non solde \\n")
    END
    ELSE
        res := compte_non_valable
    END
END
;

res ← suppression_client =
    VAR cl, bb IN
        cl ← saisir_client

```

```

; bb ← s_client_valide(cl)
; IF
    bb = TRUE
    THEN
        res := succes
    ; suppressionclient(cl)
    ; STRING_WRITE("Suppression effectuee \n")
    ELSE
        res := client_non_valable
    END
END

```

END

3.9 Composant PrmBANQUE

3.9.1 Présentation

Le composant traité dans ce paragraphe est : *PrmBANQUE*.

3.9.2 Métrique

PrmBANQUE AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/PrmBANQUE

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
Initialisation	1	0	0	0	0	100
to_Compte	2	0	0	0	0	100
to_Client	2	0	0	0	0	100
PrmBANQUE	5	0	0	0	0	100

3.9.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/PrmBANQUE.pmm non existant.

3.9.4 Source B

MACHINE

PrmBANQUE

CONSTANTS

cmi, cma, climi, clima, NOCOMPTE, NOCLIENT

PROPERTIES

cmi ∈ NAT

∧ *cma* ∈ NAT

∧ *climi* ∈ NAT

∧ *clima* ∈ NAT

∧ *cmi* < *cma*

∧ *climi* < *clima*

∧ *cmi* < *cma*

∧ *cma* > 0

∧ *cma* < MAXINT

∧ *climi* < *clima*

$\wedge \text{ clima} > 0$
 $\wedge \text{ clima} < \text{MAXINT}$
 $\wedge \text{ NOCOMPTE} = \text{cmi} \dots \text{cma}$
 $\wedge \text{ NOCLIENT} = \text{climi} \dots \text{clima}$

OPERATIONS

$cc \leftarrow \text{to_Compte} (nn) =$

PRE

$nn \in \text{INT}$

$\wedge \text{cmi} \leq nn$

$\wedge nn \leq \text{cma}$

THEN

$cc : \in \text{NOCOMPTE}$

END

;

$cl \leftarrow \text{to_Client} (nn) =$

PRE

$nn \in \text{INT}$

$\wedge \text{climi} \leq nn$

$\wedge nn \leq \text{clima}$

THEN

$cl : \in \text{NOCLIENT}$

END

END

3.10 Composant PrmBANQUE_I

3.10.1 Présentation

Le composant traité dans ce paragraphe est : PrmBANQUE_I.

3.10.2 Métrique

PrmBANQUE_I AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/PrmBANQUE_I

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	11	0	0	0	0	100
Initialisation	1	0	0	0	0	100
to_Compte	5	1	0	1	0	100
to_Client	5	1	0	1	0	100
PrmBANQUE_I	22	2	0	2	0	100

3.10.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/PrmBANQUE_I.pmm non existant.

3.10.4 Source B

IMPLEMENTATION

PrmBANQUE_I

REFINES

```

PrmBANQUE
VALUES
  cmi = 200
;  cma = 210
;  climi = 400
;  clima = 410
;  NOCOMPTE = cmi .. cma
;  NOCLIENT = climi .. clima

```

```

OPERATIONS
cc ← to_Compte (nn)=
  BEGIN
    cc := nn
  END
;
cl ← to_Client (nn)=
  BEGIN
    cl := nn
  END

```

END

3.11 Composant REQUETES

3.11.1 Présentation

Le composant traité dans ce paragraphe est : REQUETES.

3.11.2 Métrique

REQUETES AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/REQUETES.r

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
Initialisation	1	0	0	0	0	100
saisir_compte	2	0	0	0	0	100
saisir_client	2	0	0	0	0	100
saisir_somme	2	0	0	0	0	100
affiche_compte	2	0	0	0	0	100
affiche_client	2	0	0	0	0	100
REQUETES	11	0	0	0	0	100

3.11.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/REQUETES.pmm non existant.

3.11.4 Source B

```

MACHINE
  REQUETES

```

SEES

PrmBANQUE

OPERATIONS

$cc \leftarrow \text{saisir_compte} =$

BEGIN

$cc : \in \text{NOCOMPTE}$

END

;

$cc \leftarrow \text{saisir_client} =$

BEGIN

$cc : \in \text{NOCLIENT}$

END

;

$ss \leftarrow \text{saisir_somme} =$

BEGIN

$ss : \in \text{INT}$

END

;

$\text{affiche_compte}(com) =$

PRE

$com \in \text{NOCOMPTE}$

THEN

skip

END

;

$\text{affiche_client}(cli) =$

PRE

$cli \in \text{NOCLIENT}$

THEN

skip

END

END

3.12 Composant REQUETES_I

3.12.1 Présentation

Le composant traité dans ce paragraphe est : REQUETES_I.

3.12.2 Métrique

REQUETES_I AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/REQUETES_I

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	1	0	0	0	0	100
InstanciatedConstraintsLemmas	0	0	0	0	0	100
Initialisation	1	0	0	0	0	100

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
saisir_compte	10	7	0	7	0	100
saisir_client	10	7	0	7	0	100
saisir_somme	7	4	0	4	0	100
affiche_compte	6	1	0	1	0	100
affiche_client	6	1	0	1	0	100
REQUETES_I	41	20	0	20	0	100

3.12.3 Règles ajoutées à la preuve

Fichier */home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/REQUETES_I.pmm* non existant.

3.12.4 Source B

IMPLEMENTATION

REQUETES_I

REFINES

REQUETES

SEES

BASIC_IO, PrmBANQUE

IMPORTS

UTILITAIRES

OPERATIONS

```

cc ← saisir_compte =
  VAR cn IN
    BEGIN
      STRING_WRITE("Entrez un numero de compte (max 4 chiffres)\n")
      ;   cn ← INTERVAL_READ(cmi, cma)
      ;   cc ← compte_numero(cn)
    END
  END
;
cc ← saisir_client =
  VAR cl IN
    BEGIN
      STRING_WRITE("Entrez un numero de client (max 4 chiffres)\n")
      ;   cl ← INTERVAL_READ(climi, clima)
      ;   cc ← client_numero(cl)
    END
  END
;
ss ← saisir_somme =
  BEGIN
    STRING_WRITE("Entrez une somme d'argent \n")
    ;   ss ← INTERVAL_READ(1, MAXINT )
  END
;
affiche_compte (com) =

```



```

VAR icom IN
BEGIN
icom ← compteEnEntier(com)
; INT_WRITE(icom)
END
END
;
affiche_client (cli) =
VAR icom IN
BEGIN
icom ← clientEnEntier(cli)
; INT_WRITE(icom)
END
END
END

```

3.13 Composant UTILITAIRES

3.13.1 Présentation

Le composant traité dans ce paragraphe est : UTILITAIRES.

3.13.2 Métrique

UTILITAIRES AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/UTILITAIRES

	NbObv	NbPO	NbPri	NbPra	NbUn	%Pr
Initialisation	1	0	0	0	0	100
compte_numero	2	0	0	0	0	100
client_numero	2	0	0	0	0	100
compteEnEntier	2	0	0	0	0	100
clientEnEntier	2	0	0	0	0	100
UTILITAIRES	9	0	0	0	0	100

3.13.3 Règles ajoutées à la preuve

Fichier /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/UTILITAIRES.pmm non existant.

3.13.4 Source B

MACHINE

UTILITAIRES

SEES

PrmBANQUE

OPERATIONS

$cc \leftarrow \text{compte_numero}(ci) =$

PRE

$ci \in \text{INT}$

```

     $\wedge$     $ci \leq cma$ 
     $\wedge$     $cmi \leq ci$ 
THEN
     $cc \in NOCOMPTE$ 
END
;
 $cl \leftarrow \text{client\_numero}(cli) =$ 

PRE
     $cli \in \text{INT}$ 
     $\wedge$     $cli \leq clima$ 
     $\wedge$     $climi \leq cli$ 
THEN
     $cl \in NOCLIENT$ 
END
;
 $ii \leftarrow \text{compteEnEntier}(nn) =$ 

```

```

PRE
     $nn \in NOCOMPTE$ 
THEN
    ANY  $jj$  WHERE
     $jj \in \text{INT} \wedge jj \leq cma \wedge cmi \leq jj$ 
    THEN
     $ii := jj$ 
    END
END
;
 $ii \leftarrow \text{clientEnEntier}(nn) =$ 

```

```

PRE
     $nn \in NOCLIENT$ 
THEN
    ANY  $jj$  WHERE
     $jj \in \text{INT} \wedge jj \leq clima \wedge climi \leq jj$ 
    THEN
     $ii := jj$ 
    END
END
END

```

3.14 Composant UTILITAIRES_I

3.14.1 Présentation

Le composant traité dans ce paragraphe est : UTILITAIRES_I.

3.14.2 Métrique

UTILITAIRES_I AutoProved /home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/sources/UTILITA

	NbObv	NbPO	NbPRi	NbPRa	NbUn	%Pr
ValuesLemmas	1	0	0	0	0	100
Initialisation	1	0	0	0	0	100
compte_numero	5	1	0	1	0	100
client_numero	5	1	0	1	0	100
compteEnEntier	2	5	0	5	0	100
clientEnEntier	2	5	0	5	0	100
UTILITAIRES_I	16	12	0	12	0	100

3.14.3 Règles ajoutées à la preuve

Fichier */home/enseignants/attiogbe/B/Mesprojets/New1Bancaire/bdp/UTILITAIRES_I.pmm* non existant.

3.14.4 Source B

IMPLEMENTATION

UTILITAIRES_I

REFINES

UTILITAIRES

SEES

PrmBANQUE

OPERATIONS

$cc \leftarrow \text{compte_numero}(ci) =$

BEGIN

cc := ci

END

;

$cl \leftarrow \text{client_numero}(cli) =$

BEGIN

cl := cli

END

;

$ii \leftarrow \text{compteEnEntier}(nn) =$

BEGIN

ii := nn

END

;

$ii \leftarrow \text{clientEnEntier}(nn) =$

BEGIN

ii := nn

END

END

4 DICTIONNAIRE

... a ...

affiche_client

defined in REQUETES_I.imp
operation name

affiche_client

defined in REQUETES.mch
operation name
used in INTERFACE_BANQUE_I.imp (*nouveau_client*)
used in INTERFACE_BANQUE_I.imp (*nouveau_compte*)

affiche_compte

defined in REQUETES_I.imp
operation name

affiche_compte

defined in REQUETES.mch
operation name
used in INTERFACE_BANQUE_I.imp (*nouveau_client*)
used in INTERFACE_BANQUE_I.imp (*nouveau_compte*)

ancsolde

defined in BANQUE_I.imp
operation local variable
used in BANQUE_I.imp (*retrait*)
modified in BANQUE_I.imp (*retrait*)

ancsolde

defined in BANQUE_I.imp
operation local variable
used in BANQUE_I.imp (*approvisionnement*)
modified in BANQUE_I.imp (*approvisionnement*)
modified in BANQUE_I.imp (*approvisionnement*)

ancsolde1

defined in BANQUE_I.imp
operation local variable
used in BANQUE_I.imp (*transfert*)
modified in BANQUE_I.imp (*transfert*)

ancsolde2

Contents

1	INTRODUCTION	1
2	STRUCTURE GENERALE	2
2.1	Introduction	2
2.2	Graphe de dépendance	2
2.3	Métrique	2
3	DESCRIPTION DES COMPOSANTS	3
3.1	Composant APPLICATION	3
3.1.1	Présentation	3
3.1.2	Métrique	3
3.1.3	Règles ajoutées à la preuve	3
3.1.4	Source B	3
3.2	Composant APPLICATION_I	3
3.2.1	Présentation	3
3.2.2	Métrique	3
3.2.3	Règles ajoutées à la preuve	4
3.2.4	Source B	4
3.3	Composant BANQUE	6
3.3.1	Présentation	6
3.3.2	Métrique	6
3.3.3	Règles ajoutées à la preuve	6
3.3.4	Source B	6
3.4	Composant BANQUE_ETENDUE	10
3.4.1	Présentation	10
3.4.2	Métrique	10
3.4.3	Règles ajoutées à la preuve	11
3.4.4	Source B	11
3.5	Composant BANQUE_ETENDUE_I	12
3.5.1	Présentation	12
3.5.2	Métrique	12
3.5.3	Règles ajoutées à la preuve	12
3.5.4	Source B	13
3.6	Composant BANQUE_I	14
3.6.1	Présentation	14
3.6.2	Métrique	14
3.6.3	Règles ajoutées à la preuve	15
3.6.4	Source B	15
3.7	Composant INTERFACE_BANQUE	19
3.7.1	Présentation	19
3.7.2	Métrique	19
3.7.3	Règles ajoutées à la preuve	20
3.7.4	Source B	20
3.8	Composant INTERFACE_BANQUE_I	22
3.8.1	Présentation	22
3.8.2	Métrique	22
3.8.3	Règles ajoutées à la preuve	22
3.8.4	Source B	22
3.9	Composant PrmBANQUE	27
3.9.1	Présentation	27
3.9.2	Métrique	27

3.9.3	Règles ajoutées à la preuve	27
3.9.4	Source B	27
3.10	Composant PrmBANQUE_I	28
3.10.1	Présentation	28
3.10.2	Métrique	28
3.10.3	Règles ajoutées à la preuve	28
3.10.4	Source B	28
3.11	Composant REQUETES	29
3.11.1	Présentation	29
3.11.2	Métrique	29
3.11.3	Règles ajoutées à la preuve	29
3.11.4	Source B	29
3.12	Composant REQUETES_I	30
3.12.1	Présentation	30
3.12.2	Métrique	30
3.12.3	Règles ajoutées à la preuve	31
3.12.4	Source B	31
3.13	Composant UTILITAIRES	32
3.13.1	Présentation	32
3.13.2	Métrique	32
3.13.3	Règles ajoutées à la preuve	32
3.13.4	Source B	32
3.14	Composant UTILITAIRES_I	33
3.14.1	Présentation	33
3.14.2	Métrique	33
3.14.3	Règles ajoutées à la preuve	35
3.14.4	Source B	35

4 Dictionnaire 36