

An Event-B Specification of CoffeeClub

This Event-B system is based on a model that appeared in the book: System Modelling & Design Using Event-B by Ken Robinson.

It illustrates how a an abstract machine describes the consistency of a single bank account. The refinement then add multiple membership accounts while still maintaining the single bank account.

It also illustrates the use of a witness to drop (or replace) a variable when refining an event.

1	CONTEXT CoffeeClubCtx	2
1.1	MEMBER	2
2	MACHINE CoffeeClubMch	3
2.1	piggybank	3
2.2	FeedBank(<i>amount_feed</i>)	3
2.3	RobBank(<i>amount_rob</i>)	3
3	REFINEMENT CoffeeClubRef	4
3.1	accounts coffeeprice members	4
3.2	SetPrice(<i>new_price</i>)	4
3.3	NewMember(<i>new_member</i>)	4
3.4	Contribute(<i>contribution member</i>) refines FeedBank	4
3.5	BuyCoffee(<i>member_buy</i>) refines RobBank	5

CONTEXT CoffeeClubCtx

1

SETS

1.1

MEMBER

AXIOMS

ax0: finite(MEMBER)

END

VARIABLES

2.1

piggybank The coffe club has a single bank account storing all its money.

INVARIANTS

inv1: $piggybank \in \mathbb{N}$ The bank account can be positive or zero, but not negative.

EVENT **INITIALISATION**

THEN

init_0: $piggybank := 0$

END

EVENT **FeedBank**

2.2

When money is put into the bank account we feed it.

ANY

amount_feed

WHERE

grd_1: $amount_feed \in 1..100$

THEN

act_1: $piggybank := piggybank + amount_feed$

END

EVENT **RobBank**

2.3

Likewise, when taking money, we rob it.

ANY

amount_rob

WHERE

grd_1: $amount_rob \in 1..50$ The cost of a coffe is 1 up to 50.

grd_2: $amount_rob \leq piggybank$

THEN

act_1: $piggybank := piggybank - amount_rob$

END

We now introduce the concept of member accounts, the sum of the member accounts should be the total piggy bank account.

REFINES *CoffeeClubMch*
SEES *CoffeeClubCtx*

VARIABLES

members
accounts
coffeeprice

3.1

INVARIANTS

inv1_1: *members* \subseteq *MEMBER*
inv1_2: *accounts* \in *members* $\rightarrow \mathbb{N}$
inv1_3: *coffeeprice* \in 1..30

EVENT *INITIALISATION*
EXTENDS *INITIALISATION*
THEN

init1_1: *members* := \emptyset
init1_2: *accounts* := \emptyset
init1_3: *coffeeprice* := 1

END

EVENT *SetPrice*
ANY

3.2

new_price

WHERE

grd0: *new_price* \in 1..30

THEN

act0: *coffeeprice* := *new_price*

END

EVENT *NewMember*
ANY

3.3

new_member

WHERE

grd0: *new_member* \in *MEMBER*
grd1: *new_member* \notin *members*

THEN

act0: *accounts*(*new_member*) := 0
act1: *members* := *members* \cup {*new_member*}

END

EVENT *Contribute*
REFINES *FeedBank*
ANY

3.4

```

    contribution
    member
WHERE
    grd0:    contribution  $\in$  1..70
    grd1:    member  $\in$  members
    grd2:    member  $\in$  dom(accounts)
WITH
    amount_feed:    amount_feed = contribution
THEN
    act0:    accounts(member) := accounts(member) + contribution
    act1:    piggybank := piggybank + contribution
END

EVENT BuyCoffee
REFINES RobBank
ANY
    member_buy
WHERE
    grd1_1:    member_buy  $\in$  dom(accounts)
    grd1_2:    accounts(member_buy)  $\geq$  coffeeprice
    grd1_3:    coffeeprice  $\leq$  piggybank
WITH
    amount_rob:    amount_rob = coffeeprice
    The amount is replaced with the coffee price. Note that proof for amount_rob
     $\in$  1..50 is easily proven since the coffeeprice is defined as coffeeprice  $\in$  1..30
THEN
    act1_1:    accounts(member_buy) := accounts(member_buy) - coffeeprice
    act1_2:    piggybank := piggybank - coffeeprice
END

```

3.5

accounts, 4

BuyCoffee, 5

CoffeeClubCtx, 2, 4

CoffeeClubMch, 3, 4

CoffeeClubRef, 4

coffeeprice, 4

Contribute, 4

FeedBank, 3, 4

INITIALISATION, 3, 4

members, 4

NewMember, 4

piggybank, 3

RobBank, 3, 5

SetPrice, 4