

B Assignment 1

Ken Robinson

9th March 2009

Name of assignment: **ass1**

Assessment: 5 marks + 2 bonus

Submission: `~cs2111/bin/give ass1 CoffeeClub.zip`

Important: please read

This assignment is intended to help you become familiar with using Rodin, with creating projects and machine, and with entering information into the bodies of machines. This should help you become familiar with a number of functions of the Rodin toolkit. The marked up machines are appended to this specification. They are presented in the ISO characters used in the toolkit; some of those characters must be entered using ASCII representations.

Please see bonus requirement.

1 Requirement

A specification of **CoffeeClub_ctx**, **CoffeeClub** and **MemberShip** machines, as discussed in the lectures, is required. You must carry out the actions described in the following section.

2 What you have to do

Please read and follow the instructions carefully.

1. **Create a new development directory** You can call it what you want, but for the purposes of this specification, it will be called **CoffeeClub**.
2. Run Rodin toolkit and create or load a workshpace.
3. Use the Event-B explorer to create a project called *CoffeeClub*.
4. Use the Event-B explorer to create a new context machine called **CoffeeClub_ctx**.
5. Enter the context of the machine.
6. Save the machine.
7. Fix any errors and check the proof obligations in Event-B explorer.
8. Repeat the above steps for **CoffeeClub** and *Membership*.
9. Please note that the listing in this assignment is not a copy of the *Pretty Print* display under Rodin. In particular the status of an event is not shown. Status is chosen from the second drop-down menu to the right of the event name.

All the events in the machines here should be *ordinary*.

10. Export (File menu) a zip archive of the CoffeeClub project.
11. **Submit assignment** run `~cs2111/bin/give ass1 CoffeeClub.zip`.

3 Bonus

As a bonus: add comments/annotations to the components of the machines that explain the significance of the component in terms of the required functionality of the model. Important points for annotating are: *the invariants, the guards, the actions*. Less important, but could be annotated are the events and event parameters.

Annotations must document the significance item being annotated in terms of the significance to the model. The annotation should not simply paraphrase the expression being annotated. For example, the annotation of the guard

$$\textit{amount} \leq \textit{piggybank}$$

should be something like

$$\textit{ensure enough money in piggybank}$$

and not

$$\textit{amount less than or equal to piggybank}$$

4 CoffeeClub_ctx Context

CONTEXT CoffeeClub_ctx

SETS

MEMBER

AXIOMS

axm1 : *finite*(*MEMBER*)

END

5 CoffeeClub Machine

MACHINE CoffeeClub

SEES CoffeeClub.ctx

VARIABLES

piggybank

INVARIANTS

inv1 : *piggybank* $\in \mathbb{N}$

EVENTS

Initialisation

begin

act1 : *piggybank* := 0

end

Event *FeedBank* $\hat{=}$

any

amount

where

grd1 : *amount* $\in \mathbb{N}$

then

act1 : *piggybank* := *piggybank* + *amount*

end

Event *RobBank* $\hat{=}$

any

amount

where

grd1 : *amount* $\in \mathbb{N}$

grd2 : *amount* \leq *piggybank*

then

act1 : *piggybank* := *piggybank* - *amount*

end

END

6 MemberShip Machine (Refinement)

MACHINE MemberShip

REFINES CoffeeClub

SEES CoffeeClub_ctx

VARIABLES

piggybank

members

accounts

coffeeprice

INVARIANTS

inv1 : *piggybank* $\in \mathbb{N}$

inv2 : *members* $\subseteq MEMBER$

inv3 : *accounts* $\in members \rightarrow \mathbb{N}$

inv4 : *coffeeprice* $\in \mathbb{N}_1$

EVENTS

Initialisation

extended

begin

act2 : *members* := \emptyset

act3 : *accounts* := \emptyset

act4 : *coffeeprice* := $\in \mathbb{N}_1$

end

Event *SetPrice* $\hat{=}$

any

amount

where

grd1 : *amount* $\in \mathbb{N}_1$

then

act1 : *coffeeprice* := *amount*

end

Event *NewMember* $\hat{=}$

any

member

where

grd1 : *member* $\in MEMBER \setminus members$

```

    then
        act1 : members := members ∪ {member}
        act2 : accounts(member) := 0
    end

Event Contribute ≐
refines FeedBank

    any
        member
        amount
    where
        grd1 : member ∈ members
        grd2 : amount ∈ ℕ1
    then
        act1 : accounts(member) := accounts(member) + amount
        act2 : piggybank := piggybank + amount
    end

Event BuyCoffee ≐

    any
        member
    where
        grd1 : member ∈ members
        grd2 : accounts(member) ≥ coffeeprice
    then
        act1 : accounts(member) := accounts(member) − coffeeprice
    end

Event FeedBank ≐
refines FeedBank

    any
        amount
    where
        grd1 : amount ∈ ℕ
        grd2 : amount ∈ ℕ1
    then
        act1 : piggybank := piggybank + amount
    end

Event RobBank ≐

```

```
extends RobBank  
  begin  
    skip  
  end  
END
```