

# EventB Assignment 3: Travel Card Update

Ken Robinson

12th May 2008

Name of assignment: **ass3**

Due date: **18th May 2008**

Assessment: 20 marks max.

Submission: give cs2111 ass3 TravelCard.zip

This is a set of updates to resolve some issues with the travel card requirements.

## 1 Update 2

This update answers questions I've been asked, and also make some comments from experience with my own solution to the assignment.

### 1.1 Costing and Locations

*Does the cost depend on the mode of transport? Is travel from Central by bus different from travel by train?*

Yes, transport cost does depend on mode of transport, but it's not simply mode of transport. The points on a journey are not places, like *Central* or *Circular Quay*, they're nodes in a transport network. So their entry and exit points carry with them a precise identification of the actual train or bus or whatever on which the passenger is travelling.

This does not complicate your job, it makes it easier. The cost function knows how to compute the cost given the two end points.

Also discounts may relate to a previous journey and the new one being commenced, for example bus to ferry.

### 1.2 Discounts

In the last update discounting was incorporated into the cost function. This stays, and of course makes life easy for you, but it may not have been the best way of modelling it. It might have been better to have a separate function that looks for discounts. Of course, this does not affect you either way, as the computation is done remotely, but if you have a view on it you can present it in your model.

### 1.3 Exceptions

*What happens if a card is not validated on entry to the current journey?*

You don't have to handle this. Assume that a person cannot get on to the bus or train, etc without validating, and similarly cannot get out of the transport without presenting their card to a reader. These points of entry/exit can be made pretty tough. In any case breaches of the system will be handled by people, not by the travelcard itself.

If a traveller presents a card that cannot be validated then their card will not contain information about commencing the current trip.

When a card is submitted to an inspector for checking then it will be expected that the most recent record on the card records the start of a journey on this particular bus, train, etc. You can if you wish, record an infringement on the card. There will be some sort of record and presumably a possible fine.

You are, of course, free to make up your own rules if you feel you have to but these should be consistent with existing ideas.

## 1.4 Time

There is a problem with modelling *Time*, and this includes *Date*. You can have a set called *DateTime*, but the problem is modelling changing time.

Some things I've said in lectures might be interpreted incorrectly. It is true that a constant in a context machine does represent any time, but since it is a constant, it can't represent varying time.

You could have a variable that is never changed. Since it is a variable then it does represent varying time, but that's not good enough: you need continuously increasing time.

I can't see any good/simple way of modelling this. You could have an event called Tick that advances time by one. This will not be a big issue in the marking.

## 1.5 Recording events

As mentioned previously all activities with the card must be logged. In practice these will probably be logged on the card itself, and also remotely on the some travelcard centre. You don't have to differentiate between these; you can regard this as part of the implementation. Clearly, updated information must be part of a machine state. Context machine constant state can be used to model information that is essentially static for the purpose of any event.

## 1.6 A Prover Problem

Since you are probably using a second order function for cost you will hit an annoying problem with the prover.

Suppose you have a function  $F : X \rightarrow (Y \rightarrow Z)$

First notice for what follows that it will be very useful to make the functions total, rather than partial, so try to do that.

Even so, when you use something like  $F(x)(y)$

You will get POs of the form:

$x : \text{dom}(F)$

and a bit later

$y : \text{dom}(f(x))$

which, unfortunately recurses back to a repetition of  $x : \text{dom}(F)$ !!!

Anyhow, the way to discharge  $x : \text{dom}(F)$  is to enter

$\text{dom}(F) = X$

in the Proof Control area

and then to select "ah" (add hypothesis) in the Proof Control menu bar.

This should be proved easily using the (green) auto prover, but you will need to click on it.

You can see why it was important to use total functions; otherwise you don't have a ready hypothesis for  $\text{dom}(F)$  or  $\text{dom}(F(x))$ .

This becomes a real pain and I will happily allow a blue review solution, but only for total functions.

## 2 Update 1

### 2.1 Journey costing

A journey is defined by two points: the *start* of the journey and the *end* of the journey. You may assume that these end points also uniquely identify the type of transport. The cost of a journey will be given by a single function call  $cost(cardid, start, end)$ . This may not be the same as  $cost(cardid, end, start)$ , see *Discounting*. This function should be specified in a context machine, as should the set of journey start/end points.

### 2.2 Card balance

For the purpose of this assignment we will assume that the balance on a card can become negative, but that the balance at the start of a journey must be positive.

Given that the card is anonymous this presents a problem of debt recovery, but we will ignore this for this assignment.

### 2.3 Card validation

At the beginning of a journey the card is validated. Validation consists of checking that the card is valid and that the cash balance on the card is sufficiently positive. The minimum cash balance should be specified in a context machine, and may depend on the starting point.

### 2.4 Discounting

It can be assumed that discounting is automatic and is a function of the cardid. So, for example, there may be a discount on a ferry journey following a bus journey, provided that the ferry journey occurs within some maximum time interval after the end of the bus journey. The discount will be determined automatically when the card is validated at the start of the ferry trip.

### 2.5 Logging

All journeys must be logged. This will be done at the start and end of each journey.