Available at the COMP2111 website.

Assignment 1 COMP2111 13s1
Cleaners to the Rescue

Kai Engelhardt

Revision: 1.3 of Date: 2013/04/09 14:47:11

This assignment is worth 14 marks and due before the end of week 5, that is, **Sunday April 14th, 23:59:59** local time Sydney.

Problem Statement

The Hotel Association has become aware of a serious flaw in their members' most commonly deployed door lock and key card system. Disappointed with the manufacturer, they hire you for an independent modelling and assessment of (a) the existing system and (b) a potential improvement suggested by a moderately inebriated scientist K who observed that rooms are routinely cleaned after a guest checks out and before the next guest can check in.

Original System

The original lock system stores one key in the re-codable door lock and two on each card. If either key matches then the door opens. If the first key matches then the key stored in the lock is overwritten with the second key on the card. Reception creates a new card whenever a guest checks in. Such a new card contains the current key of the door to the room allocated to the guest as first key and a fresh key as second key. Cleaners access rooms using cards with a single master key (and no second key) that is hard-coded into door locks and different to all keys any guests could have on their cards.

Proposed Improvement

In the new and improved system as proposed by K there are 2 master keys: one for regular cleaning and one for thorough cleaning after check-out. Cleaners carry two cards holding a single master each and use them according to their cleaning tasks. Lock behaviour is changed such that the first key on a guest card only opens the door when the immediately preceding access was by a cleaner using the second master key. Otherwise the second key has to match the stored key in the lock.

Tasks

The first 3 tasks are Event-B modeling and proving tasks to be carried out in Rodin. The last task is a documentation task.

Task 1: Abstract Model

Model the room access problem abstractly, that is, without keys, cards, and locks, just by maintaining room bookings and occupancy of rooms. Relevant events are `CheckIn`, `CheckOut`, `Enter`, `Exit`, and `Clean`. The first two affect room bookings, the second pair affects occupancy and cleanliness of rooms, the last only establishes cleanliness of a room.

Formulate essential invariants to guarantee that no guest can access a room he has not booked.

Task 2: Original System

Model the original lock system faithfully. Feel free to do some research. Even if you find something online, it will have to be adapted significantly before it counts as a serious attempt. Use the animator to demonstrate potential flaws in the system. Set the model up as a refinement of the abstract model. It is to be expected that some POs cannot be discharged unless additional and perhaps unrealistic invariants are added to this model. What is the weakest invariant you can add to establish refinement?

Task 3: Proposed Improvement

Model K's suggestion as an alternative refinement of the abstract model. Can you prove refinement? If not, what is a counter example to K's claim that his idea works?

Task 4: Documentation

Describe and justify your modelling decisions in a LaTeX document named `HotelKeys.tex`. Make it concise, convincing, and fun to read for your tutor. Don't include lengthy Event-B listings.

Deliverables

`HotelKeys.zip` is your Event-B project exported from `Rodin`.

`HotelKeys.tex` is a LaTeX document with your name or student number mentioned in the `\author` command. It contains your report.

Submission Instructions

The `give` command to be run is:

```
% 2111
% give cs2111 ass1 HotelKeys.zip HotelKeys.tex
```

The command above submits the bare minimum. Should you feel the need to include more files, e.g., for vector diagrams, just list them as well.