Project Proposals

Introduction

Purpose of this document

In this document you will find a series of projects that are proposed to you. Each project is introduced in the form of a short text which is not particularly well written as a "requirement document": this is done on purpose as we want you to be aware of what you will encounter in Industry where such documents are usually rather poor.

If you don't like the proposed project given in this document, you can propose a project of your own with a text similar to the ones below. We shall decide whether you can proceed with such a project.

Team and Timing

Except for two projects that are to be done individually (by one student only), all other projects are to be done by little teams of students not exceeding three of you. Different teams can chose the same project. The choice of a project and the forming of the team can be done as soon as possible. Please, give your team decision to Liu Haiyang. The work on each project can start immediately once the choice is done and this until mid-November.

Your Task

Your task is divided into a series of phases. This corresponds to what has been shown many times in the course:

1. Write a proper requirement document.
2. Propose a refinement strategy.
3. Develop the project by constructing various more and more refined models.
4. Show your final result in a short oral presentation (2min).

At the end of the first two phases, you are asked to hand out a document containing your proposals for a requirement document and for your refinement strategy. This is the *first milestone*. It can be around October 13th.

In the middle of phase 3, you'll have a *second milestone* where you will show your intermediate formal development result. This second milestone might occur around October 27th.

Note that the previous dates are provisional: they can be modified if necessary.

Your formal development has to be performed with the Rodin Platform. Don't hesitate to have many refinement steps, introducing gradually functionalities and equipment assumptions. Don't forget to have many comments in your models: explain how you follow your development strategy and how you take care of your requirements.

The models should be proved with the Rodin Platform. If you have problems with the prover, try to understand why and record this in a separate document.

Don't hesitate to ask for *help* if needed.

Project 1: Cash Machine

A cash machine is a device allowing customers to get cash by introducing their Credit Card into it.

A cash machine contains a certain quantity of money. More money is put regularly within the machine. When money is put in the machine, it cannot be used by customers.

The credit card of a customer contains a PIN number. When introducing the card in the fence of the Cash Machine, the user is asked to enter his Pin Number. In case the customer enters a wrong number, the machine allows for 2 more trials. After more additional negative trials, the card is swallowed by the machine.

If the card is accepted then the machine asks for the quantity of money to be removed (predefined quantity correspond to different buttons to be depressed). The Credit Card of each customer contains a maximum allowance for each week. In other words, people cannot get more cash than a certain quantity each week. If the machine has not enough money to give to the user, then the customer cannot be served.

The credit card contains the bank name and account number of the customer. The machine checks that the customer has enough money on his account before delivering the cash.

If the demand of the customer is accepted, then the machine delivers the money only after the customer has removed his card.

An early removal of its credit card by the customer stops the transaction.

Project 2: Automatic Doors of a Subway

This project consists in modeling the behavior of the automatic doors of modern subways as one can see them in the Beijing subway.

There are two different doors: the doors of the platform and the doors of the train.

Some safety properties imposes that the train can only move when both categories of doors are closed. Conversely, the two categories of doors can be opened only when the train is stationnary.

Study carefully (in the subway) the relative synchronization between the two categories of doors.

Envisage some incidents (people blocking a door) and the way they can be analyzed and fixed by the people in charge (put these people and their behavior in your model)

Project 3: Analysis of a Continuous Signal (for a single student)

The purpose of this exercise is to construct the model of a system analysing a continuous signal in order to transform it into a a "step" signal.

In figure 1, you can see a continuous input signal being sampled every other CT seconds (CT stands for the Cycling Time): this is indicated by the black dots. An output signal will be generated as a result of the sampling. Initially, the output signal is off.

If the sampling detects that the threshold RTH (Rising Threshold) has been passed between two successive samplings and that the input signal is above RTH for a time BT (deBounce Time) immediately after this detection, then the output signal moves from off to on. We have a symmetric situation with the threshold FTH (Falling Threshold) and the output signal moving from on to off. We suppose that the integer ratio $n = BT/CT$ is well defined and positive.

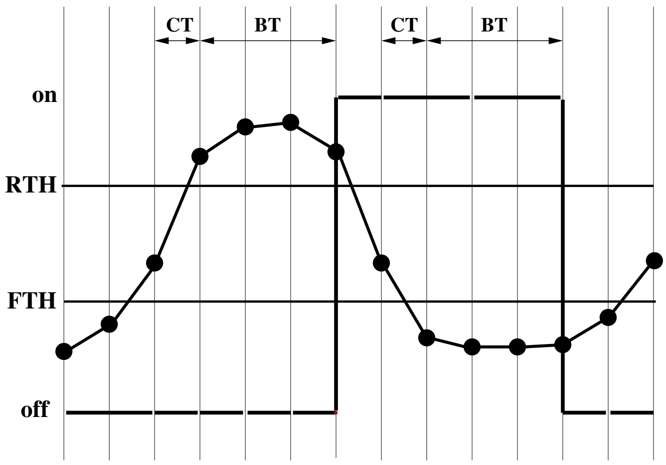

Fig. 1. Sampling

Construct a model of this system producing an adequate output for each detected input. Each event will correspond to a reaction of the system to an input. There will be many different events as there are many different situations in which an input may occur.

Define your model by successive refinements. Don't forget to write an initial requirement document as well as a refinement strategy before embarking into the formal development.

Project 4: An Hotel Electronic Key System

The purpose of this project is to develop the model of an hotel electronic key system.

The purpose of such a system is to guarantee that between your check-in and check-out in an hotel, you can enter the room you booked and no one else can do so. Note that this is not the case with a metallic key system since a previous user of the room may have duplicated the metallic key.

A proposed implementation is defined as follows:

(1) Each hotel room door is equipped with an independent electronic lock which holds an electronic key. The lock has a fence in which one may insert a magnetic card.

(2) Each check-in starts a new booking of a certain room. To each booking is associated a magnetic card containing two electronic keys: a guaranteed new key, and the electronic key presently stored in the lock of the room (a centralised dynamic systems is supposed to hold the keys stored in the room locks). For entering the room, you insert your card in the fence of the lock. The lock reads your card and opens the door provided its own electronic key is one of the keys in your card. The lock electronic key is replaced by the new key which is in your card.

In figure 2 a new card is introduced in the fence. It contains the new key $k2$ and the key $k1$ of the lock. The card is accepted and the electronic key of the lock becomes $k2$. The owner of the card can now re-enter his room with the same card (since it contains key $k2$).

The proposed implementation with the cards requires that people effectively use the room. If someone does not use the room he booked then the next client cannot enter the room.

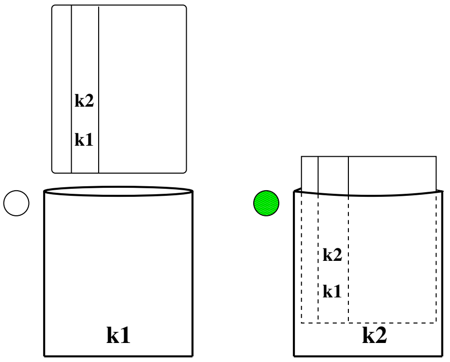

Fig. 2. A Door Lock and a New Card Being Inserted

Develop a model for this system. Do not introduce the card and the key at the beginning. You rather make an abstraction were the main property is expressed: a person who has booked a room is guaranteed that no one else can enter this room.

In subsequent refinements, express the fact that clients are served according to their arrival (hotel policy). Then finally, introduce the card system which implement this policy.

You might define various events: check-in, check-out, enter_room, leave_room. Consider also a master entering in the room (under the responsibility of the hotel).

Project 5: Almost Linear Sorting (for a single student)

Normally the expected time to sort n items is proportional to $n \log n$. But, in certain circumstances, it can be made "almost" proportional to n. The purpose of this project is to develop the model of such an almost linear sorting algorithm.

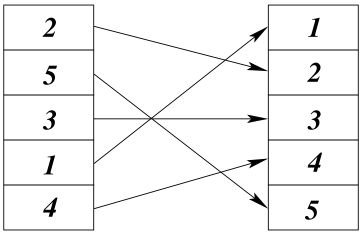

Fig. 3. Linear Sorting

Suppose we have to sort n distinct numbers ranging exactly from 1 to n. The sorting is clearly linear: simply put each number i at the ith position. This is illustrated in figure 3.

Now suppose we have to sort n distinct number ranging from 1 to m where m is slightly greater than n. For instance, n is equal to 5 but m is now equal to 7 as shown in figure 4. We can suppose that this assumption about a slight difference only between m and n could help designing an almost linear sorting.

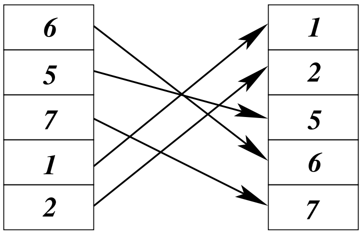

Fig. 4. Almost Linear Sorting

This is so because there is no reason indeed for that small difference between n and m to suddenly induce a large difference in the sorting time with respect to the linear time we had when n and m were identical.

First define a context introducing n, m and the array f to sort:

$$n > 0$$

$$m > 0$$

$$f \in 1\,..\,n \rightarrowtail 1\,..\,m$$

Define an initial machine doing the sorting in one shot. For this, define a variable array g which is a function from $1\,..\,n$ to $1\,..\,m$. Also define an event sort and an anticipating event progress as follows:

```
sort
  any $h$ where
    $h \in 1\,..\,n \rightarrow 1\,..\,m$
    $\mathrm{ran}(h) = \mathrm{ran}(f)$
    $\forall i \cdot i \in 1\,..\,n-1 \;\Rightarrow\; h(i) < h(i+1)$
  then
    $g := h$
  end
```

```
progress
  status
    anticipated
  begin
    $g :\in 1\,..\,n \rightarrow 1\,..\,m$
  end
```

Refine the initial machine. For this, introduce a variable k situated in $0\,..\,m$ and a variable l situated in $0\,..\,n$. Initially, these variables are set to 0. As an invariant, state that the array g is sorted from 1 to l. Also state that the image $g[1\,..\,l]$ is equal to $\mathrm{ran}(f) \cap 1\,..\,k$, and that the cardinal of $\mathrm{dom}(f \rhd 1\,..\,k))$ is exactly equal to l. Refine the events. Split the anticipated event progress: the two resulting events become convergent. Prove this refinement.

Add another refinement with an event scan constructing a boolean array r from $1\,..\,m$ to BOOL where we eventually have $r(x)$ being TRUE if and only if x is in the range of f. Refine the events. Prove this refinement. Perform an animation. The sorting is proportional to $m + n$, so roughly $2n$ when n and m are almost equal.

Project 6: Lift

The informal description of an elevator system is given below. It's quite clumsy and poorly written. Sometimes, some basic requirements might be omitted (when they have been considered trivial). This is on purpose. It reflects the average user's requirements document one encounters in practice.

The elevator system consists of the following parts:

- an elevator,
- a door for the elevator,
- a cable and an engine for moving the elevator,
- sensors for detecting that the elevator has reached some floor,
- an engine for opening and closing the door,
- sensors for detecting if the door is open or closed,
- $N + 1$ floors,
- buttons on floors for calling the elevator,
- buttons in the elevator for choosing floors,
- a controller that controls the system,
- copper wires between them.

In order for the user to know that his request is acknowledged by the system, all buttons have a small light attached to them. That light should be switched on when the user presses the corresponding button. Conversely, it should be switched off once the request has been served.

Floors have two buttons (one for each direction of the elevator), unless only one button is needed. There are exactly $N + 1$ buttons in the elevator (one for each floor).

Finally, to prevent accidents, the elevator should always move with the door engine working towards closing the doors: this is to prevent users from opening the doors while the elevator is moving.

The inputs of the controller are:

- the status of the cable engine (winding, unwinding or stopped),
- the status of the door engine (opening, closing or stopped),
- the status of the floor sensors (the number of the floor that the elevator has reached or -1 if the elevator is between two floors; floors are counted from 0 to N),
- the status of the door sensors (fully open, half open or closed),
- the status of the buttons (pressed or not: boolean).

The outputs of the controller are:

- the command of the cable engine (wind, unwind or stop),
- the command of the door engine (open, close or stop),
- the command of the lights of the buttons (on or off: boolean).

Out of that description, you should write a clean requirements document. You should use the following taxonomy of requirements:

- EQP for equipment,
- FUN for functional,
- SAF for safety requirements.

Pursue this project by proposing a refinement strategy and then develop the corresponding model by means of several refinements.

Project 7: A Business Protocol

This projects aims at constructing the model of a business protocol. The idea is also to use the *design pattern* technique which was presented in the Press example.

This protocol determines the negotiation taking place between a buyer and a seller. The outcome of the protocol might be as follows:

- the two parties agree on a final agreement by which the seller sells a certain quantity of a certain product to the buyer at a certain price. Note that the product, the quantity, and the price are all abstracted here as an INFO exchanged between the participants.
- the two parties might end up by not succeeding in finding an agreement,
- whatever the final result (agreement or no agreement), the buyer might always cancel the protocol.

The protocol is divided up into four phases: the *initial phase*, the *free game* phase, the *last proposal* phase, and the *termination* phase..

- In the initial phase, the buyer starts the protocol by sending a proposal to the seller.
- After this initial proposal has been received by the seller, the protocol enters the free game phase. In this second phase buyer and seller can send counter-proposal or acceptance to the other partner proposal in a fully asynchronous way. In this phase, an acceptance or a counter-proposal by either party is never definitive.
- The last proposal phase is at the initiative of the buyer which makes it clear to the seller that the proposal sent to it is the last one: the seller can either accept it or reject it. It cannot send a counter-proposal.
- The termination phase is the one by which the buyer sends a termination message which the seller has to acknowledge.

During the three first phases, the buyer can always cancel the protocol by sending a message to the seller which needs to acknowledge it. This has the immediate effect to move the protocol to the termination phase.

When the seller or the buyer sends a counter-proposal it must mention in the corresponding message to which proposal of the other party it corresponds.

The channels between the seller and the buyer are not reliable: messages can be lost, copied and do not arrive necessarily at their destination in the same order in which they have been send.

Use design patterns to handle the sending of messages, the response to a message or both. Use these patterns in a systematic fashion to model the various phases of the protocol.

Do not perform the modelling in a flat manner: use various refinements to structure your formal model. Do not forget to write a precise requirement document as well as a refinement strategy.