Projet 3A IL
Année académique 2026/2027
Date de remise 1 février 2027

Dominique Méry

23 novembre 2026@9:52pm

Table des matières

1 Consignes **2**

2 Etapes du projet **2**

3 Etudes de cas **2**

3.1 Sujets pour groupe de deux personnes 2
3.1.1 Sujet 1 Calcul de la fonction lightpolynome 2
3.1.2 Sujet 2 Calcul de la fonction x^6 2
3.1.3 Sujet 3 Calcul de la fonction primes 3
3.1.4 Sujet 4 Calcul de la fonction perfect 3
3.1.5 Sujet 5 Calcul de la fonction summation 3
3.1.6 Sujet 6 Calcul de la fonction searchmatrix 3
3.1.7 Sujet 7 Calcul de la fonction closure 3
3.1.8 Sujet 8 Calcul de la fonction x^7 4
3.1.9 Sujet 9 Quicksort 4
3.2 Sujets pour groupe de trois personnes 6
3.2.1 Sujet 1 Surveillance des températures (sujet à 3) 6
3.2.2 Sujet 2 Le Passage Automatisé Rapide aux Frontières Extérieures (PARAFE) (sujet à 3) 6
3.3 Sujets pour groupe de quatre personnes 7
3.3.1 Sujet 3 Gestion des clés électroniques des chambres d'un hôtel 7
3.3.2 Sujet 4 Contrôle d'accès au GLORIA 9
3.3.3 Sujet 5 Services en ligne 9
3.3.4 Sujet 6 The access card SUICA 10
3.3.5 Sujet 7 Portillons d'accès 12
3.3.6 Sujet 8 Gestion des autoroutes 14

1 Consignes

Le projet est à réaliser au sein d'un groupe de 2,3 ou 4 élèves. Chaque groupe rendra un dossier électronique sous la forme d'une archive zip dont le nom sera de la forme `groupe-x-nom.zip` où x est le numéro du groupe et nom est le nom d'un des membres du groupe.
Chaque sujet peut être choisi par au plus un groupe.

2 Etapes du projet

10 janvier 2027 Remise d'un document court (`groupe-x-nom-doc1.pdf`) reprenant l'étude de cas choisie et précisant les propriétés de correction (sûreté ou safety) attendues du modèle à construire.

1 février 2027 Date de remise du dossier final (`groupe-x-nom.zip`) : 1 février 2027 à dominique.mery@loria.fr et sujet `projettelecom`

3 Etudes de cas

3.1 Sujets pour groupe de deux personnes

Pour les problèmes ci-dessous, on demande de modéliser les éléments mathématiques nécessaires notamment les suites permettant de caractériser la solution. De plus, il faut utiliser la patron de conception formelle appelé *iterative pattern* pour construire la chaine de machines selon la relation de raffinement. A partir de ce développement, il faut construire un algorithme et le vérifier avec Frama-c en ajoutant les pré- et post-conditions et les invariants nécessaires pour déduire automatiquement la correction partielle et l'absence d'erreurs à l'exécution. En particulier, il faudra utiliser les machines Event-B pour déduire les invariants. Le dossier devra comporter :

— Un document court reprenant le problème choisi et précisant les propriétés de correction (invariants, pre/post, rte) attendues du modèle à construire.
— Un dossier d'analyse et de présentation de l'étude menée : une présentation synthétique des modèles, indication des choix faits, des difficultés rencontrées, des invariants, des propriétés de sûreté
— l'archive du projet Rodin développé.
— le programme Frama-c construit à partir des modèles

3.1.1 Sujet 1 Calcul de la fonction lightpolynome

La fonction $lightpolynome$ calcule la valeur $a{\cdot}n^3{+}b{\cdot}n^2{+}c$ pour tout entier naturel $n \in \mathbb{N}$ et n'utilise que des opérations d'addition. Le modèle Event-B devra donc développer par raffinement un algorithme séquentiel calculant cette fonction. Pour réaliser ce développement, il faudra définir les suites permettant de calculer cette fonction. Il faudra appliquer les règles de tranformation pour dériver un algorithme C analysable par Frama-c ; Le développement servira aussi de mine d'informations pour produire un invariant et une preuve du contrat en Frama-c.

3.1.2 Sujet 2 Calcul de la fonction x^6

La fonction f_6 est définie comme suit : $\forall n \in \mathbb{N}.f_6(n) = n^6$. On dispose d'une seule opération arithmétique pour la calculer et celle-ci est l'addition. En revanche, on connaît des identités remarquables qui développe $(a{+}b)^6$ et on peut utiliser cette identité, pour construire un ensemble de suites récurrentes qui permettent de calculer $(n{+}1)^6$ en fonction de n^6. Pour ce projet, on souhaite construire un algorithme qui calcule n^6 en utilisant uniquement des fonctions d'addition. On pourra s'inspirer du calcul des fonctions n^2 et n^3 pour trouver l'ensemble des suites récurrentes nécessaires pour calculer n^6. Il faudra aussi vérifier le programme C

produit à partir des modèles Event-B, en utilisant le logiciel *Frama-c*. Il faudra de plus écrire une version annotée le programme C avec les bonnes assertions.

3.1.3 Sujet 3 Calcul de la fonction primes

La fonction $primes$ renvoie une valeur vraie ou fausse selon que le nombre n est premier ou non. Un nombre n est premier, s'il a deux diviseurs distincts. Il faut donc construire une suite $d \in 0..n \rightarrow \mathbb{N}$ qui renvoie les diviseurs de n plus petits qu'une valeur i : $d(i)$ est l'ensemble des diviseurs de n plus petits que i. Le modèle Event-B devra donc développer par raffinement un algorithme séquentiel calculant cette fonction. Pour réaliser ce développement, il faudra définir les suites permettant de calculer cette fonction. Il faudra appliquer les règles de tranformation pour dériver un algorithme C analysable par Frama-c ; Le développement servira aussi de mine d'informations pour produire un invariant et une preuve du contrat en Frama-c.

3.1.4 Sujet 4 Calcul de la fonction perfect

La fonction $perfect$ renvoie une valeur vraie ou fausse selon que le nombre n est parfait ou non. Un nombre n est parfait, si la somme de ses diviseurs différents de lui est égal à ce nombre. Par exemple, 28 est parfait. On pourra construire une suite $p \in 0..n \rightarrow \mathbb{N}$: $p(i)$ est l'ensemble des diviseurs de n plus petits que i. Il faudra aussi définbir une suite calcuylabnt la somme des diviseurs. Le modèle Event-B devra donc développer par raffinement un algorithme séquentiel calculant cette fonction. Pour réaliser ce développement, il faudra définir les suites permettant de calculer cette fonction. Il faudra appliquer les règles de tranformation pour dériver un algorithme C analysable par Frama-c ; Le développement servira aussi de mine d'informations pour produire un invariant et une preuve du contrat en Frama-c.

3.1.5 Sujet 5 Calcul de la fonction summation

La fonction $summation$ calcule la valeur $t[1]^2+t[2]^2+\ldots+t[n]^2$ pour un entier naturel $n \in \mathbb{N}$ et n'utilise que des opérations d'addition. Le modèle Event-B devra donc développer par raffinement un algorithme séquentiel calculant cette fonction. Pour réaliser ce développement, il faudra définir les suites permettant de calculer cette fonction. Il faudra appliquer les règles de tranformation pour dériver un algorithme C analysable par Frama-c ; Le développement servira aussi de mine d'informations pour produire un invariant et une preuve du contrat en Frama-c.

3.1.6 Sujet 6 Calcul de la fonction searchmatrix

La fonction $searchmatrix$ renvoie deux valeurs une valeur booléenne et une valeur scalaire de type $int{\cdot}int$ et ses paramètres sont une matrice $M[n,m]$ d'entiers int et une valeur val de type int :

- $searchmatrix(M,val) = (T,(i,j))$ si, et seulement si, $M[i,j] = val$
- $searchmatrix(M,val) = (F,(i,j))$ si, et seulement si, $\forall(k,l).(k,l)in1..n{\times}1..m \Rightarrow M[k,l] \neq val$

Le modèle Event-B devra donc développer par raffinement un algorithme séquentiel calculant cette fonction. Pour réaliser ce développement, il faudra définir les suites permettant de calculer cette fonction. Il faudra appliquer les règles de tranformation pour dériver un algorithme C analysable par Frama-c ; Le développement servira aussi de mine d'informations pour produire un invariant et une preuve du contrat en Frama-c.

3.1.7 Sujet 7 Calcul de la fonction closure

La fonction $closure$ renvoie une valeur CM matrice de booléens et ses paramètres sont une matrice $M[n,n]$ de booléens : $\forall(k,l).(k,l)in1..n{\times}1..n$

- $CM[k,l] = T$ si, et seulement si, il existe un chemin de k à l pour M.
- $CM[k,l] = F$ si, et seulement si, il n'existe pas de chemin de k à l pour M.

CM est définie comme la somme des produits M^i pour i de 0 à n. On doit dopnc définir le produit de deux matrices et la somme de matrices.
Le modèle Event-B devra donc développer par raffinement un algorithme séquentiel calculant cette fonction. Pour réaliser ce développement, il faudra définir les suites permettant de calculer cette fonction. Il faudra appliquer les règles de tranformation pour dériver un algorithme C analysable par Frama-c ; Le développement servira aussi de mine d'informations pour produire un invariant et une preuve du contrat en Frama-c.

3.1.8 Sujet 8 Calcul de la fonction x^7

La fonction f_6 est définie comme suit : $\forall n \in \mathbb{N}.f_7(n) = n^7$. On dispose d'une seule opération arithmétique pour la calculer et celle-ci est l'addition. En revanche, on connaît des identités remarquables qui développe $(a+b)^7$ et on peut utiliser cette identité, pour construire un ensemble de suites récurrentes qui permettent de calculer $(n+1)^7$ en fonction de n^7. Pour ce projet, on souhaite construire un algorithme qui calcule n^7 en utilisant uniquement des fonctions d'addition. On pourra s'inspirer du calcul des fonctions n^2 et n^3 pour trouver l'ensemble des suites récurrentes nécessaires pour calculer n^7. Il faudra aussi vérifier le programme C produit à partir des modèles Event-B, en utilisant le logiciel *Frama-c*. Il faudra de plus écrire une version annotée le programme C avec les bonnes assertions.

3.1.9 Sujet 9 Quicksort

Tony Hoare[**?**] a proposé un algorithme connu sous le nom de *quicksort* et cet algorithme a deux versions. Une version récursive et une version itérative figurent dans le programme donné dans la suite. Le problème à résoudre est de définir une suite qui permet de de simuler la version récursive sous une forme itérative. On demande de, développer la solution itérative en utilisant la suite que vous aurez mise en évidence.

```c
#include <stdio.h>


// Utility function to print an array
void printArray(int arr[], int size) {
      for (int i = 0; i < size; i++)
        printf("%2d ", i);
    printf("\n");
    for (int i = 0; i < size; i++)
        printf("%2d ", arr[i]);
    printf("\n");
}


// Function to partition the array and return the pivot index
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            // Swap arr[i] and arr[j]
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
```

```
    // Swap arr[i+1] and arr[high] (pivot)
    int temp = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp;

    return i + 1;
}

// Iterative quicksort using a stack
void iterativeQuicksort(int arr[], int low, int high) {
    int stack[high - low + 1];
    int top = -1;
    int thesize;


    // Push initial values of low and high to the stack
    stack[++top] = low;
    stack[++top] = high;

    // Keep popping from the stack until it is empty
    while (top >= 0) {
        // Pop high and low
        high = stack[top--];
        low = stack[top--];
        //    thesize = sizeof(arr) / sizeof(arr[0]);
        // Set pivot element at its correct position in the sorted array
        int p = partition(arr, low, high);
        // If there are elements on the left side of the pivot, push them to the st
        if (p - 1 > low) {
            stack[++top] = low;
            stack[++top] = p - 1;
        }

        // If there are elements on the right side of the pivot, push them to the s
        if (p + 1 < high) {
            stack[++top] = p + 1;
            stack[++top] = high;
        }
    }
}

// recursive version
void quicksort(int array[], int low, int high) {
  if (low < high) {

    // find the pivot element such that
    // elements smaller than pivot are on left of pivot
    // elements greater than pivot are on right of pivot
    int pi = partition(array, low, high);

    // recursive call on the left of pivot
    quicksort(array, low, pi - 1);
    // recursive call on the right of pivot
    quicksort(array, pi + 1, high);
  }
}
```

```
// Driver program to test the above functions
int main() {
  int arr[] = {64, 25, 12, 67, 22, 12, 11};
    int size = sizeof(arr) / sizeof(arr[0]);

    printf("Sorted array: \n");
    printArray(arr, size);
    iterativeQuicksort(arr, 0, size - 1);
    printf("Sorted array: \n");
    printArray(arr, size);

    return 0;

}
```

3.2 Sujets pour groupe de trois personnes

- Un document court reprenant l'étude de cas choisie et précisant les propriétés de correction (sûreté ou safety) attendues du modèle à construire.
- Un dossier d'analyse et de présentation de l'étude menée : une présentation synthétique des modèles, indication des choix faits, des difficultés rencontrées, des invariants, des propriétés de sûreté
- l'archive du projet Rodin développé.
- un petit programme de démonstration construit à partir des modèles

Les sujets ci-dessous sont donnés dans la suite :

3.2.1 Sujet 1 Surveillance des températures (sujet à 3)

Un dispositif est constitué d'un capteur de température et de pression C et d'un réseau de communication R autorisant des échanges entre une application Aconnectée au réseau et le capteur. Une application permet de calculer la moyenne des températures collectées par un capteur placé dans un lieu donné. Cette application dispose aussi de fonctionnalités diverses notamment celle de produire une alerte si la température collectée vérifie une condition donnée comme par exemple la température est inférieure à 0 degres Celsius. Cette application reçoit des mesures selon une régularité choisie par l'utilisateur via l'application. Pour résumer, les fonctions de l'application sont :

- acquisition des températures/pressions.
- calcul de la moyenne des températures/pressions reçues
- détection d'un événement attendu sur une température et/ou sur une pression
- réglages de paramètres.

La modélisation devra tenir compte des erreurs possibles des données transmises ou encore du choix des unités (Celsius, Kelvin, Fahrenheit).

3.2.2 Sujet 2 Le Passage Automatisé Rapide aux Frontières Extérieures (PARAFE) (sujet à 3)

Le système PARAFE est mis en place par les services de contrôle des frontières notamment à l'aéroport CDG. Ce système comprend les équipements suivants :

- un tunnel de franchissement avec deux portes
- un lecteur pour la passeport situé à l'extérieur et contrôlant la porte d'accès dans le tunnel
- un lecteur contrôlant la sortie de ce tunnel et contrôlant la porte de sortie de ce tunnel ; ce lecteur lit un élément biologique comme une empreinte ou un iris.



FIGURE 1 – Parafe 1



FIGURE 2 – Parafe 2

Le sas (voir les figures 1 et 2) est un couloir qui comprend deux portes et deux lecteurs. Un des lecteurs est à l'extérieur et l'autre à l'intérieur. Il y a un mécanisme qui permet de savoir s'il n'y a qu'une seule personne dans le sas.
Le protocole de franchissement (voir la figure 3) de ce tunnel ou sas est le suivant :

- Le passeport est lu et authentifié par un lecteur à l'entrée dans le sas
- Le sas est ouvert et la personne peut ainsi y entrer.
- Le sas se referme et le personne doit alors présenter un doigt
- La seconde porte s'ouvre et la personne peut quitter le sas.
- Le sas se referme et la perzsonne suivante peut procéder à sa sortie.

3.3 Sujets pour groupe de quatre personnes

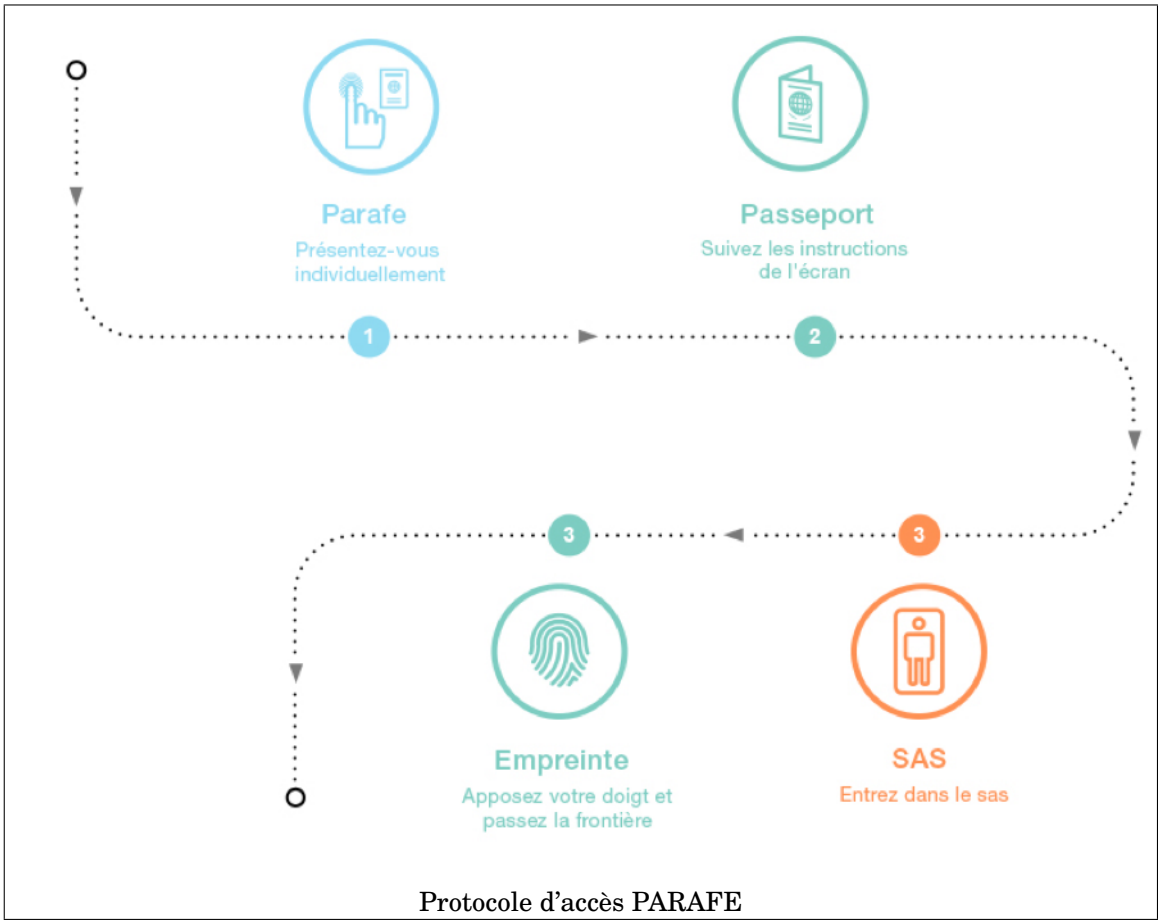
Pour les sujets ci-dessous, le dossier contiendra les éléments suivants :

- Un document court reprenant l'étude de cas choisie et précisant les propriétés de correction (sûreté ou safety) attendues du modèle à construire.
- Un dossier d'analyse et de présentation de l'étude menée : une présentation synthétique des modèles, indication des choix faits, des difficultés rencontrées, des invariants, des propriétés de sûreté
- l'archive du projet Rodin développé.
- un petit programme de démonstration construit à partir des modèles

Les sujets ci-dessous sont donnés dans la suite :

3.3.1 Sujet 3 Gestion des clés électroniques des chambres d'un hôtel

Un hôtel souhaite gérer les accès à ses chambres en installant des serrures électroniques contrôlées par des cartes électroniques. Lorsqu'un client arrive à l'hôtel, il récupère une carte électronique qui est activée pour lui donner accès à sa chambre. Cette carte donne accès à une seule chambre et aucune autre carte ne donne accès à cette chambre. Lorsqu'un client arrive, il est attendu à partir de 16h00 et lorsqu'il quitte l'hôtel, il rend sa carte avant midi. Il se peut néanmoins qu'un client ne rende pas sa carte avant midi et dans ce cas, cette chambre ne peut



Protocole d'accès PARAFE

FIGURE 3 – Protocole Parafe

pas être allouée à une autre personne puisqu'il faut faire la ménage avant de la proposer à un client nouveau. Il existe une carte universelle qui ouvre toutes les portes à tout moment pour assurer la sécurité des personnes qui pourraient être blessées ou quin pourraient avoir perdu leur carte. Cependant, une carte perdue peut toujours être clonée pour permettre à une personne de poursuivre l'occupation de cette chambre. La gestion des cartes est gérée par un système informatique disposant d'un lecteur/écrivain de cartes magnétiques qui permet de les annuler, de les programmer ou de les reprogrammer. Ce système permet aussi de contrôler les serrures électroniques et de connaître l'état de la chambre (occupée,non-occupée,disponible, non-disponible). En particulier, toute chambre disponible ne pourra être attribuée qu'à au plus un client. Enfin, la remise de la carte conduira à l'établissement d'une facture qui sera acquittée par le client.

3.3.2 Sujet 4 Contrôle d'accès au GLORIA

Un laboratoire célèbre GLORIA développe des recerches top secrètes et hautement sensibles. Cette situation lui impose de contrôler l'accès de ses visiteurs et de ses employés. Pour cela, il dispose d'une brigade d'hôtesses qui peuvent, après vérification de l'identité d'un visiteur ou autre, donner un accès au bâtiùent du GLORIA. Un second accès se fait pare un mécanisme d'identifiaction fiable permettant à une personne accréditée d'accéder au bâtiment. Le mécanisme d'identification automatique pêut faire appel à des saisies d'empreintes ou à des mécanismes de sécurité maximale.
On a donc plusieurs protocoles possibles :

- Une personne A vient visiter une personne B et demande à l'hôtesse ses papiers et l'hôtesse enregistre l'identité de la personne et lui donne l'accès le cas échéant.
- Une personne A a une carte d'accès et l'utilise lors de son accès ; si l'accès lui est accordée, alors le dispositif d'accès est ouvert et la personne A peut entrer dans le bâtiment.
- Une personne A veut sortir du bâtiment et l'hôtesse est présente ; l'hôtesse enregistre la sortie de cette personne qui sort du bâtiment.
- Une personne A veut sortir du bâtiment et l'hôtesse n'est plus présente ; A doit indiquer un code sur un dispositif contrôlant la sortie.
- Une personne a une accès par carte et doit présenter cette carte pour avoir l'autorisation de sortir.

Dans la mesure où une personne peut perdre son accès quand elle est dans le bâtiment, il faut prévoir un numéro d'appel de la sécurité assurée à distance 24/245.
En cas d'urgence, les dispositifs de sécurité libèrent tous les accès pour garantir la sûreté des personnes
Les droits d'accès des personnes sont fonction de leur catégorie et selon la catégorie d'une personne, celle-ci est autorisée à entrer au laboratoire à certains moments de la journée ou des jours ouvrables ou de certains jours de congé. Personne n'est autorisé à entrer au laboratoire avant 7h00 du matin et après 22h00. Le laboratoire comprebnd deux types de personnes : les permanents et les non-permanents. Un non-permanent n'a pas le droit d'entrer après 18h00 et avant 8h00 du matin et un non-permanent n'a pas le droit d'entrer au laboratoire le samedi, le dimache ou pendant un jour férié. Quand le laboratoire est fermé, personne ne peut y pénétrer et le laboratoire est fermé à certaines dates définies par la maintenance.

- Développer une suite de modèles par raffinement en expliquant pour chaque étape les choix faits et comment ils peuvent être justifiés.
- A partir du modèle précédent, écrire un petit programme prototypant ce modèle.

3.3.3 Sujet 5 Services en ligne

Un ensemble de *services* est offert par des *fournisseurs* et chaque service est facturé au client concerné selon les tarifs du fournisseur. Chaque client doit en principe payer un accompte avant que le service ne soit assuré et une facture définitive est envoyée au client quand le service a été rendu. Un service donné peut être indisponible à certains moments de l'année et donc l'accompte doit être remboursé. Au cours de l'année, la tarification évolue et peut être modifiée même au cours d'une commande.

Il est important de noter que les fournisseurs fournissent des services facturés dans une monnaie donnée et que cette monnaie n'est pas nécessairement celle des clients. Ce point mérite votre attention dans l'analyse de ce problème.
Il est important de donner des propriétés de sûreté raisonnables.
On ne se préoccupe pas des questions de sécurité des accès qui sont assurés par d'autres moyens.

— Développer une suite de modèles par raffinement en expliquant pour chaque étape les choix faits et comment ils peuvent être justifiés.
— A partir du modèle précédent, écrire un petit programme prototypant ce modèle.

3.3.4 Sujet 6 The access card SUICA

What is SUICA? The Suica is a prepaid e-money card for moving around and shopping. There is no more need to buy a ticket from a vending machine. Just touch your Suica to the ticket gate and the fare is automatically deducted from your Suica. The Suica can be used not only for JR East trains, but subways and buses as well. See the Suica Map for transportation systems you can use with the Suica.
The Suica can also be used to pay for things with e-money. Buy soft drinks and coffee from vending machines and on the train. Even buy a newspaper at the station kiosk without fiddling for coins. For traveling in Japan, the Suica makes a more pleasant trip.

Riding Trains Simply touch the Suica card to the reader at the ticket gate when out and about. There is no need to purchase a ticket from a vending machine. The fare is automatically calculated at the station where you get off. When you get off the train, the fare will be calculated at that station. If the balance is insufficient, however, you will need to add more money to your Suica.
Additional tickets must be purchased to use the Suica for travel on a limited express, express or Green Car. The Suica cannot be used for travel on the Shinkansen. Valid Areas and Transportation Systems
The Suica can be used on JR East lines in the Tokyo metropolitan area as well as for subways, buses and the Tokyo Monorail that connects Haneda Airport with Tokyo.
In addition to the Tokyo area, the Suica can be used for certain transportation systems in the Sendai and Niigata, Hokkaido, Tokai, West Japan and Kyushu areas.

Recharge the Suica The Suica can be loaded and used as many times as desired.
The Suica can be loaded up to a maximum of 20,000 yen at Automatic Ticket Vending Machines and Fare Adjustment Machines displaying the Suica mark.
Only yen may be used to load a card. A credit card cannot be used to load a Suica.

Networks and companies There are severall companies and networks and when you travel, your route mau requyire to change the network anbd you have to retouch the access point.

Complementary Informations The access card SUICA is described on the website of the japanese companies and on french websites. Please consult these sites to get more informations.

Work to do The work to do is to provide an Event-B model of the system controling the access to the metro lines of Tokyo. If you read the informations, you will understand that there are severall companies and that, when travelling in the metro, you can change from a company to anbother one after having used the card which is recording your travel cost. The goal is to build a small system ablme to simulate the ccess control system called SUICA.



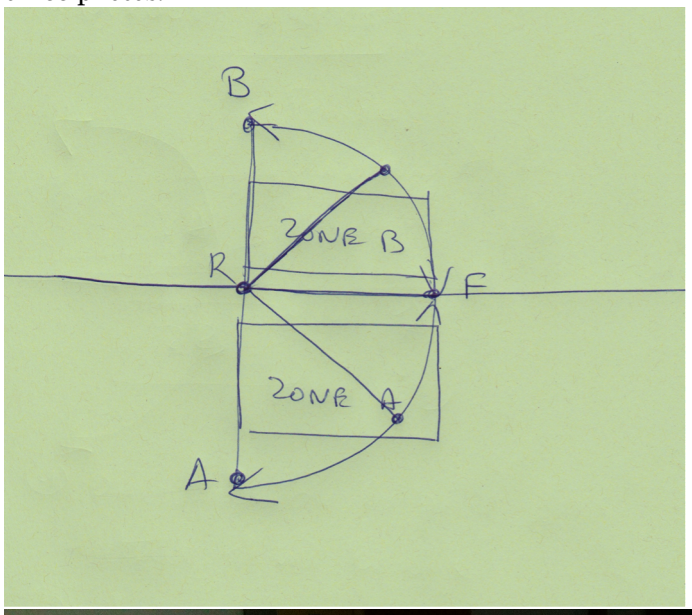
FIGURE 4 – Card SIUCA



FIGURE 5 – Portique d'accès et point de contact carte

3.3.5 Sujet 7 Portillons d'accès

A system is controling the opening and the closing of a door. The door is either closed or open. When the door is open, it is open either in the zone A or in the zone B. A zone is an area which is equipped with a sensor and which is providing to the system the information that somebody is in front of the door and that the door can not be opened in the same area but should be opened in the opposite area. The zone is used for ensuring that the door will not hurt somebody. The next diagram is sketching the real system, which is shown on the next three photos.







The door is closed in the position F and it is fully open either in the position A or the position B. The diagram gives a view of the door from the upper side. The two zones A and B (the two zones are the carpets in the photos) have a sensor able to detect that something or somebody is in.

As the door is controling the access to a specific zone, there are still rules for the control of the opening. When the request for the door is done from the zone A, then the person who wants to get in B, should first provide an access code or to introduce his/her access card into the card reader. However, when the person wants to have an access from the zone B, the system is automatically opening the door.

— Analyse the problem and identify safety requirements of the target system.
  Identify the components of the target system.
— Develop a complete modelling chain of the target system.

3.3.6 Sujet 8 Gestion des autoroutes

Le système que nous souhaitons modéliser est le système de péage des autoroutes (voir la figure 6. L'objectif est de construire un système chargé d'assurer le péage par les usagers du coût de l'utilisation de l'autoroute. L'évaluation du coût est calculée à l'aide du ticket présenté par chaque usager ou bien encore à l'aide du dispositif de télépéage. Lorsqu'un automobiliste arrive à un péage, soit il entre sur l'autoroute, soit il en sort. La figure 7 décrit le portique de péage et le dispositif à utiliser pour régler le coût, en insérant le ticket dans l'orifice prévu et en réglant le coût soit pas chèque soit par de la monnaie.
Deux protocoles sont possibles :

Ticket S'il veut entrer sur l'autoroute, il appuie sur un bouton, le portique lui délivre un ticket qu'il conserve et qui lui permet d'avoir l'ouverture de la barrière et, enfin, il franchit le portique et entre sur l'autoroute. Le portique se referme dès que le véhicule a franchi le péage. La figure 6 montre les signaux indiquant que la voie peut être utilisée dans le cas où un ticket est remis à l'automobiliste.S'il veut sortir, il insère son ticket et règle la somme demandée (voir la figure 7) ; puis le portique s'ouvre et se referme dès que le véhicule a franchi le portique. Dans ce cas, on distinguera le réglement en sortie à un employé présent ou bine le réglement par carte bancaire en sortie.

Télépéage S'il veut entrer sur l'autoroute, il emprunte un portique indiquant qu'il est destiné et réservé au télépéage. Le portique n'est pas fermé et le véhicule est détecté à son passge du portique par un dispositif spécifique. Le dispositif lit l'identité du véhicule et enregistre la date d'entrée. La figure 6 montre les signaux indiquant que la voie peut être utilisée dans le cas où un ticket est remis à l'automobiliste. S'il veut sortir, il emprunte une voie dédiée à ce type d'accès.

On suppose que deux véhicules ne peuvent pas franchir le portique ouvert simultanément. Le système doit assurer le service de manière à ce que toute personne qui emprunte l'autoroute paie la redevance due. Il est important de discuter les cas particuliers quand un automobiliste a perdu le ticket ou bien encore quand il ne peut pas régler la somme. Quelques comportaments peuvent être possibles et doivent être considérés dans votre modémisation. Si un véhicule franchit la barrière du télépéage sans disposer de la carte de télépéage, un message est envoyé aux -services de sécurité de l'autoroute et la photographie de la plaque d'immatriculation est fournie. Il est important aussi d'envisager des cas particuliers où le véhicule ne sort pas l'autorouté après un délai donné et de prendre une décision à ce sujet comme prévenir les services d'urgence.



FIGURE 6 – Vue de péage et des portes





FIGURE 7 – Vue proche d'une entrée ou sortie de péage