

Formal Modeling of the ‘Via Verde’ system in VDM++

Diogo Vaz Nunes

Gabriel Cardoso Candal

Miguel Jorge Gonçalves Pereira

Mestrado Integrado em Engenharia Informática e Computação

Métodos Formais em Engenharia de Software

17/12/2014

Contents

[1. Informal system description and list of requirements 3](#_Toc406519310)

[1.1 Informal system description 3](#_Toc406519311)

[1.2 List of requirements 3](#_Toc406519312)

[2. Visual UML model 4](#_Toc406519313)

[2.1 Use case model 4](#_Toc406519314)

[2.3 Class model 4](#_Toc406519315)

[3. Formal VDM++ model 5](#_Toc406519316)

[3.1 Class ViaVerde 5](#_Toc406519317)

[4. Model validation 5](#_Toc406519318)

[4.1 Class MyTestCase 5](#_Toc406519319)

[4.2 Class TestViaVerde 6](#_Toc406519320)

[5. Model verification 6](#_Toc406519321)

[5.1 Example of domain verification 6](#_Toc406519322)

[5.2 Example of invariant verification 6](#_Toc406519323)

[6. Conclusions 6](#_Toc406519324)

[7. References 6](#_Toc406519325)

# 1. Informal system description and list of requirements

## 1.1 Informal system description

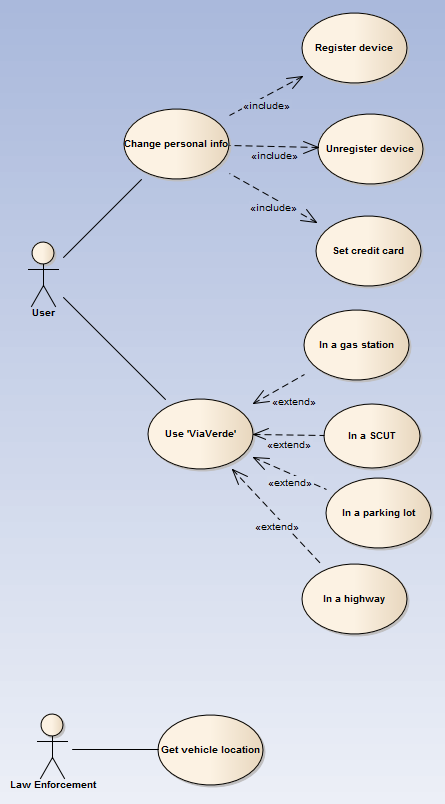
Via Verde facilitates the usage and payment of certain services through the attachment of a device to the front of a vehicle. Those services include highways, parking lots, gas stations and SCUT. For those vehicles not identified by a device, the system will record its usage anyway, so that the user can associate a device to that vehicle when he wishes.

## 1.2 List of requirements

|  |  |  |
| --- | --- | --- |
| **Id** | **Priority** | **Description** |
| R1 | Mandatory | Clients may register a Via Verde device for their vehicles. |
| R2 | Mandatory | Clients may unregister a Via Verde device from their previously registered vehicles. |
| R3 | Mandatory | Clients may change the credit card number associated to one of his Via Verde devices. |
| R4 | Mandatory | Clients may use the services provided by Via Verde, having a device or not. |
| R5 | Mandatory | Clients must be able to check the records for a given car. |
| R6 | Mandatory | The police must be able to retrieve a vehicle’s current location, if that vehicle is still in a park or in a highway. |

# 2. Visual UML model

## 2.1 Use case model



## 2.3 Class model

# 3. Formal VDM++ model

## 3.1 Class ViaVerde

# 4. Model validation

## 4.1 Class MyTestCase

**class** MyTestCase

/\*

Superclass for test classes, simpler but more practical than VDMUnit`TestCase.

For proper use, you have to do: New -> Add VDM Library -> IO.

JPF, FEUP, MFES, 2014/15.

\*/

**operations**

-- Simulates assertion checking by reducing it to pre-condition checking.

-- If 'arg' does not hold, a pre-condition violation will be signaled.

**protected** assertTrue: **bool** ==> ()

assertTrue(arg) ==

return

**pre** arg;

-- Simulates assertion checking by reducing it to post-condition checking.

-- If values are not equal, prints a message in the console and generates

-- a post-conditions violation.

**protected** assertEqual: ? \* ? ==> ()

assertEqual(expected, actual) ==

if expected <> actual **then** (

IO`print("Actual value (");

IO`print(actual);

IO`print(") different from expected (");

IO`print(expected);

IO`println(")\n")

)

**post** expected = actual

**end** MyTestCase

## 4.2 Class TestViaVerde

# 5. Model verification

## 5.1 Example of domain verification

## 5.2 Example of invariant verification

# 6. Conclusions

# 7. References

1. VDM-10 Language Manual, Peter Gorm Larsen et al, Overture Technical Report Series No. TR-001, March 2014
2. MFES course’s VDM++ handouts, Ana Paiva/João Pascoal Faria