Flu Epidemic

Gabriela CAVALCANTE DA SILVA, Raquel LOPES DE OLIVEIRA, Xiaohan HUANG

Université Nice Sophia Antipolis

gabicavalcantesilva, oliveira.raquel.lopes, stephanie1028han**@gmail.com**

1. Introduction

This project is a simulation of a simplification of real-world epidemics. The simulation consists of both a disease model describing characteristics of the illness, and a social model describing how people interact with each other [1].

2. Project

We have 5 states implemented:

Healthy this state represents people healthy. They can become sick after come in contact with people or animals who are contagious. But this will depends on the infection rate of the virus.

Sick this state represents the people after contact with a contagious person/animal.

Contagious this state happen after pass the incubation time and depends on the specific virus of infection. Is this state, the person or animal can infect other beings. People that are contagious or sick will move slowly.

Recovering after becoming contagious a person can starts recovering, or dies. A recovering person is no longer contagious and becomes healthy again after a few more days. After recovering this person become immune disease.

Dead depending on the mortality rate of the disease, a person may recover or die.

The states are made with the idea of ”state machine”. We have triggers that when ”fired”, the state change. Some situations can change the cicly, like a person that is vaccinated or has resistance.

In the class *StatesManagerAnimal* we ensure that the animal change his state in each case. At first we use the function *getState* to check the current state, then judge if the state should be changed in one specific case. When the animal is **sick**, if the time of infection is greater than the incubation time of the virus, the state will be changed to **contagious**. In the similar way, the state will be changed to **dead** when the contagious time greater than which of the virus. And when there are contagious animals around of a healthy animal, it will infect with the virus and his state will be changed to **dead**. But just duck and chicken can infect each other, and pigs can infect only pigs.

3. Class Diagram

Below there is a simplified class diagram about the project:

[1] http://moodle.i3s.unice.fr/mod/assign/view.php?id=4640

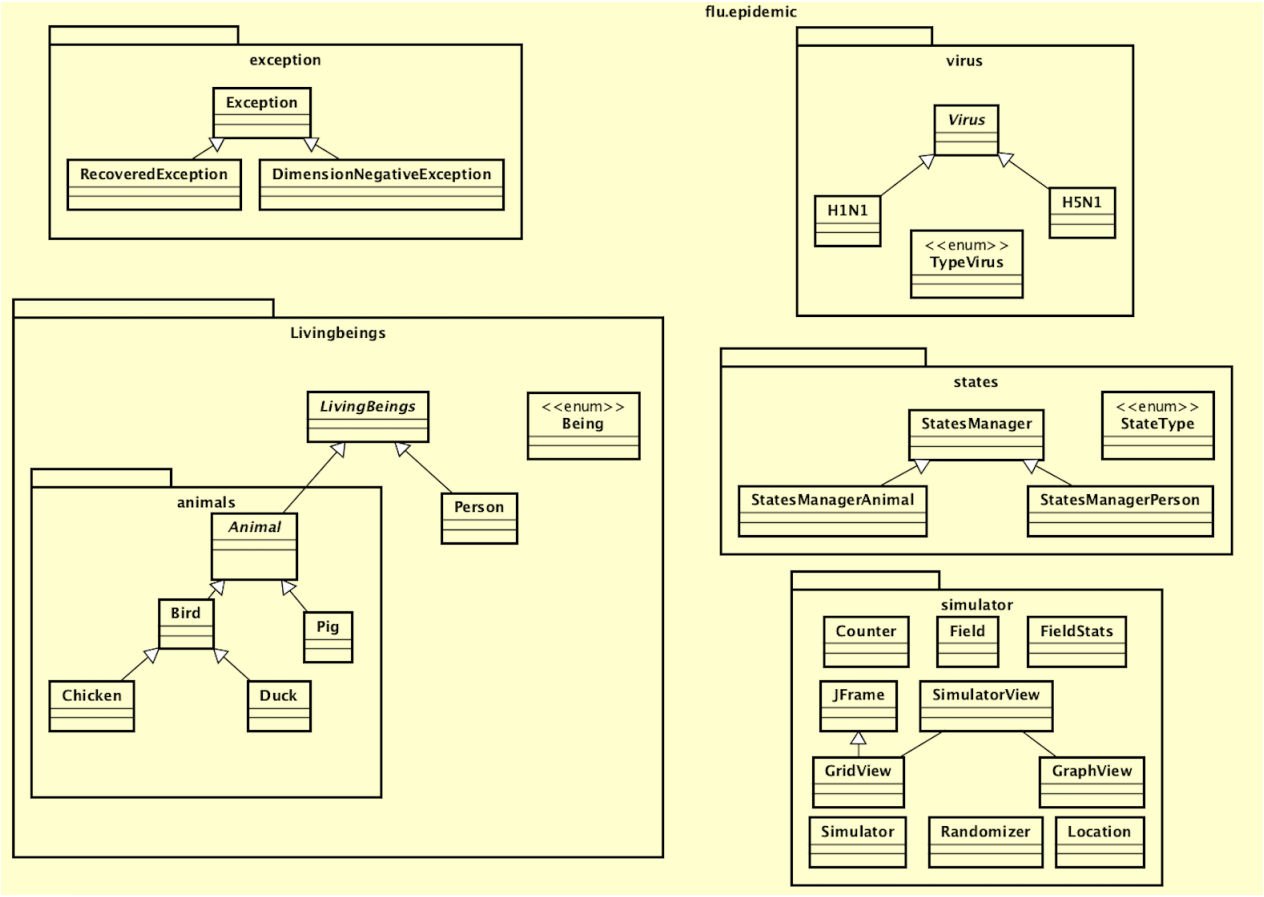

Figure 1. Class Diagram

4. Client Constraints

4.1. Explicit inheritance

As shown in the class diagram we use the Inheritance in several places. Chicken and Duck inherits Bird; Bird and Pig inherits Animal; Animal and Peson inherits LivingBeings; H1N1 and H5N1 inherits Virus; StatesManagerAnimal and StatesManager inherits StatesManager. We have to consider the ineritance works for ”grand parents” as Chicken also inherits LivingBeings for example.

4.2. Polymorphism

We use polymorphism in attributs and methods. You can verify in the code we chosed put some attributs as protected to it can addapt the way he behave (also because it will be never inicialized as a parent, so there is no problem to not be private). The same to the methods; the methods with polymorphism change the way the behave depending of the object ’used’.

4.3. Enum

We used 4 enum to help us:

```
public enum Being {
    PERSON("person"),
```

```
    CHICKEN("chicken"),
    DUCK("duck"),
    PIG("pig");

        [...]
}
```

The enum Being refers to the lowest cases of being than can happen. Realise we didn't put *Animal; Bird* because if it's an Animal it will be obligatory Chicken or Duck or Pig; if it is a Bird it will be obligatory a chicken or a duck.

```
public enum TypeVirus {
    H1N1("H1N1"),
    H5N1("H5N1");

    [...]
}
```

```
public enum StateType {
    HEALTHY("healthy"),
    SICK("sick"),
    CONTAGIOUS("contagious"),
    RECOVERING("recovering"),
    DEAD("dead");

    [...]
}
```

Theses 2 finals enums are really important to to manage the project.

5. Variants

- If a person start to recovering, our variant says it will mandatorily become a Healthy Person. There is no possibility to the "recover" don't work.
- also, there is the attribute **resistance**. If a person has been exposed and has recovered, she will be resistance to infection, and if this person touch in other person, she will "pass" the resistance.
- And the **VACCINATED RATE** that is used when a person is Healthy and can be vaccinated. In this case, this person don't become sick.

...