AGH
University of Science and Technology

Faculty of Mechanical Engineering and Robotics

Software engineering

Project: Egg Incubator

Marek Machnicki (270063)
Filip Jawień (270058)
Przemysław Dudys (279723)

Aim of project

Aim of project was to model incubator system in rhapsody, based on UML language. The system consists of heater, fan and peristaltic pump, controlled by sensors.

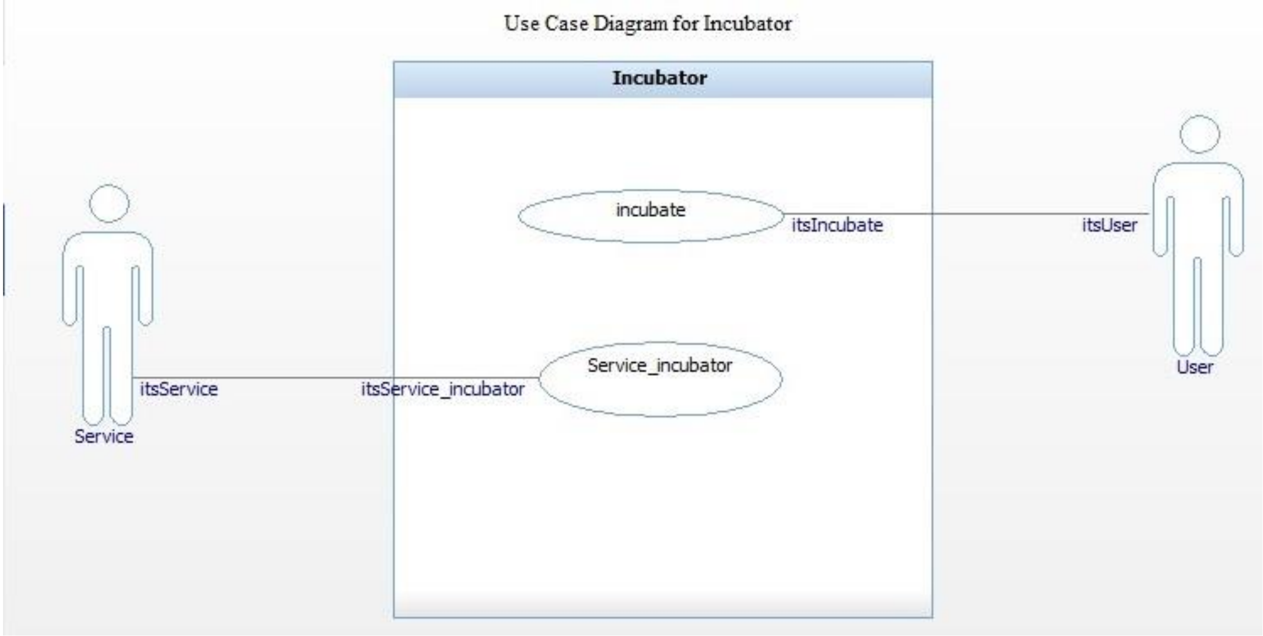
Working principle:

System is designed to keep constant temperature and humidity. At the beginning of the program, we assume that temperature is in correct range. When temperature drops below required range (simulated by user writing ‘c’ command), heater is on and after 2 seconds fan starts to blow warm air into incubator. When temperature is on correct level (simulated by user clicking ‘h’ command) heating and blowing warm air stops and incubation continues.

Simiarly when humidity drops below required value, which is simulated by user clicking ‘n’ letter, pump starts to pump for defined amount of time.

Diagram of usecases:

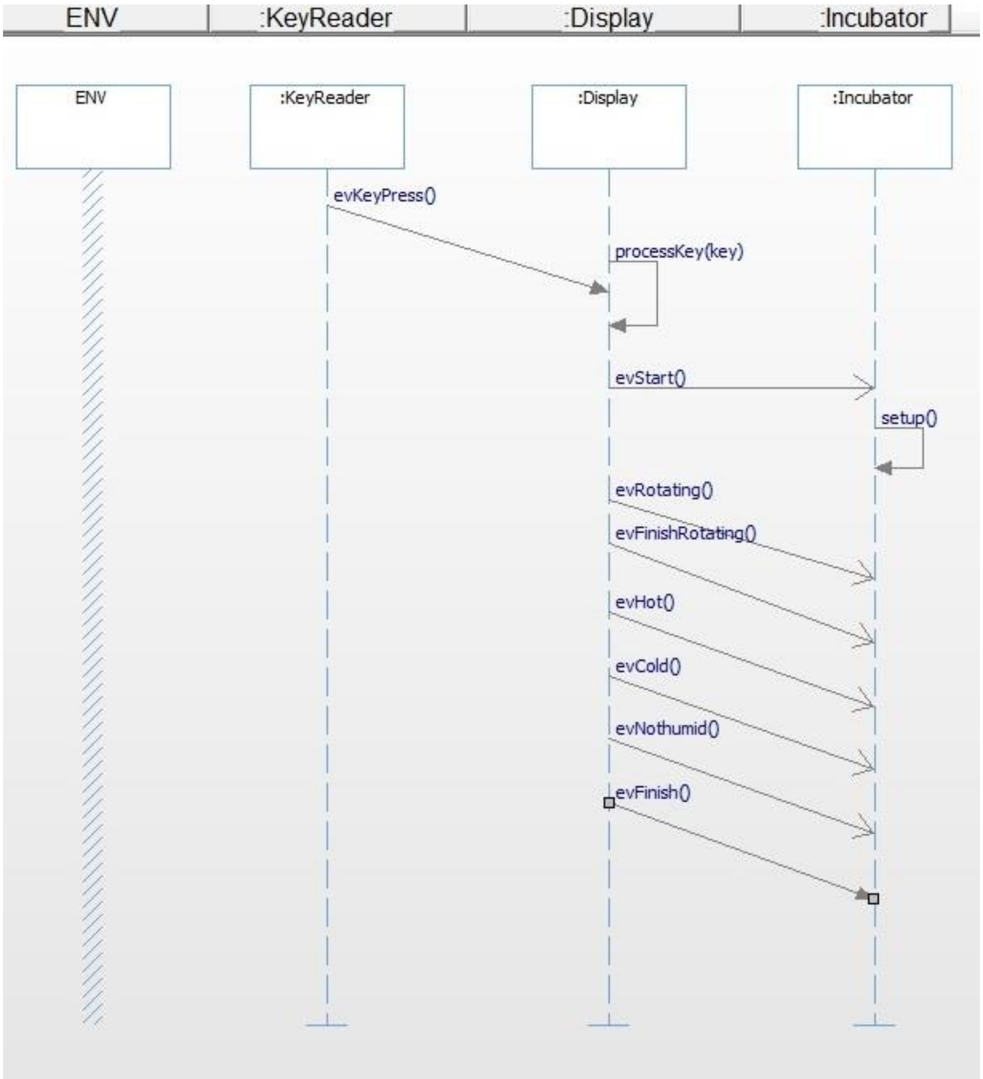
Use case diagram shows 2 actors, a User and a Service and also use cases and relations between them. The User can press the button that can press the button that starts incubation. The use case "incubate" starts process of incubation.



The sequence diagram:

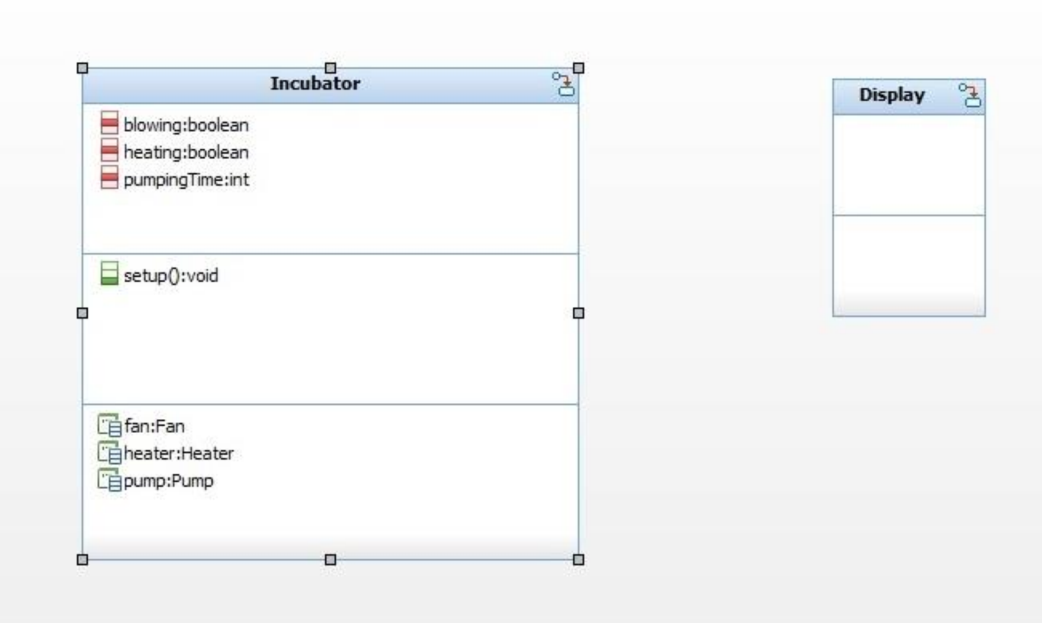
First, we can notice Display and Incubator classes. The klass display, uses keyReader class, responsible for reading data from keyboard. Then if we start incubation we wait for it to finish, or in case of any disturbances, such as low temperature or humidity (found by senors, simulated by user).

When sensor find disturbances it starts with heating and blowing, or initiated peristaltic pump for increasing humidity.



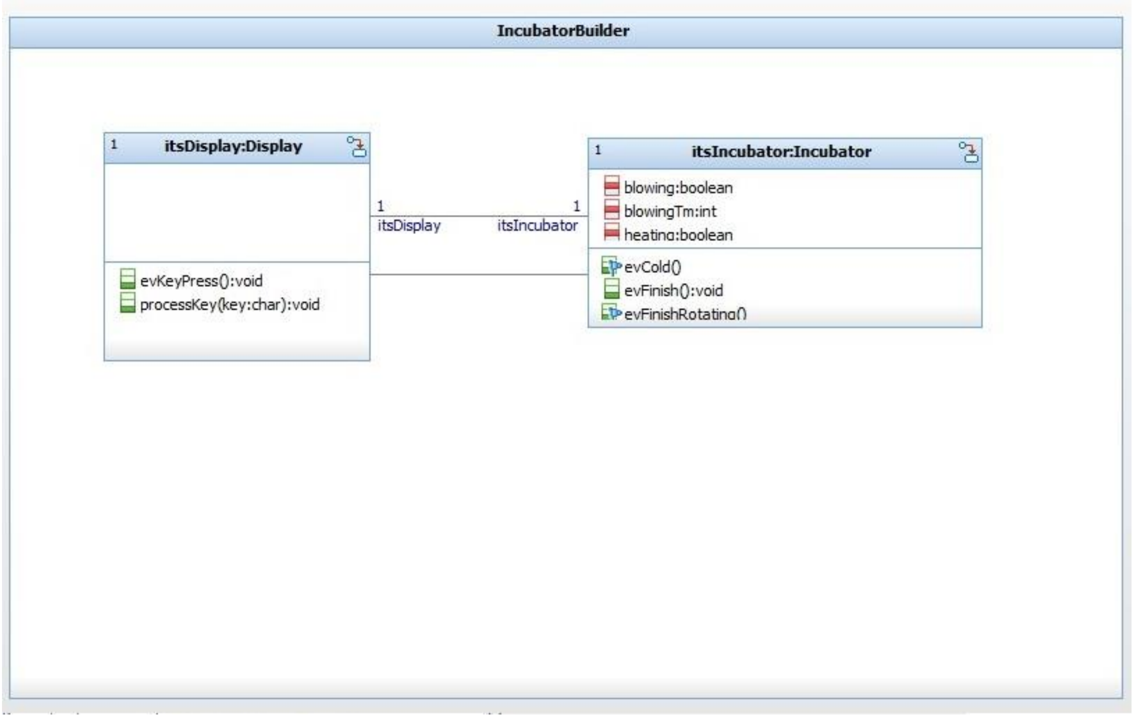
The class diagram:

Class diagram shows statics of system. Two classes are defined: “Incubator” and “Display”. The features of class incubator consists of 3 attributes of Boolean and int type, where all of them are public. Class also has 3 operations where all have public visibility.



The object diagrams:

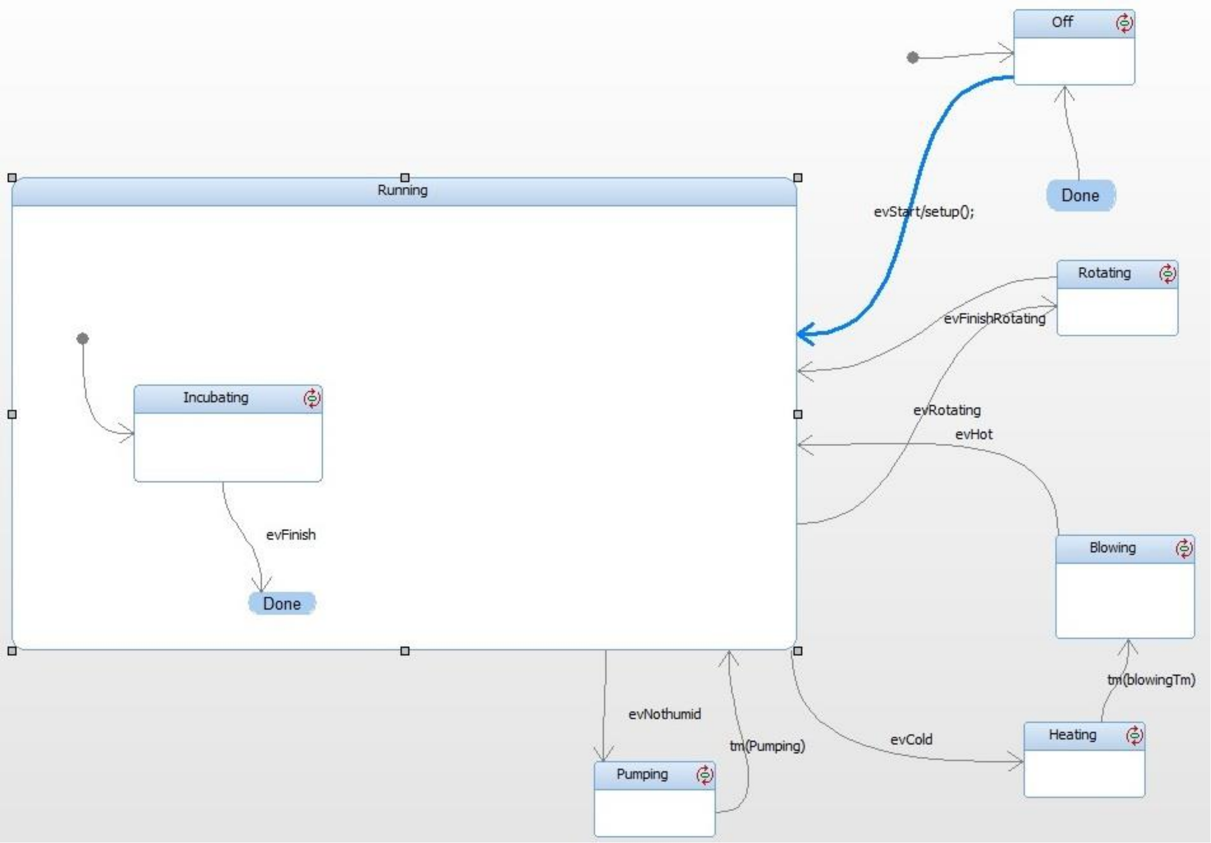
The object diagram, shows objects in the system and their correlations. They are used to define meaning of class by showing object and its connections. Here, the object are connected by binary association and are exchanging signals between each other.



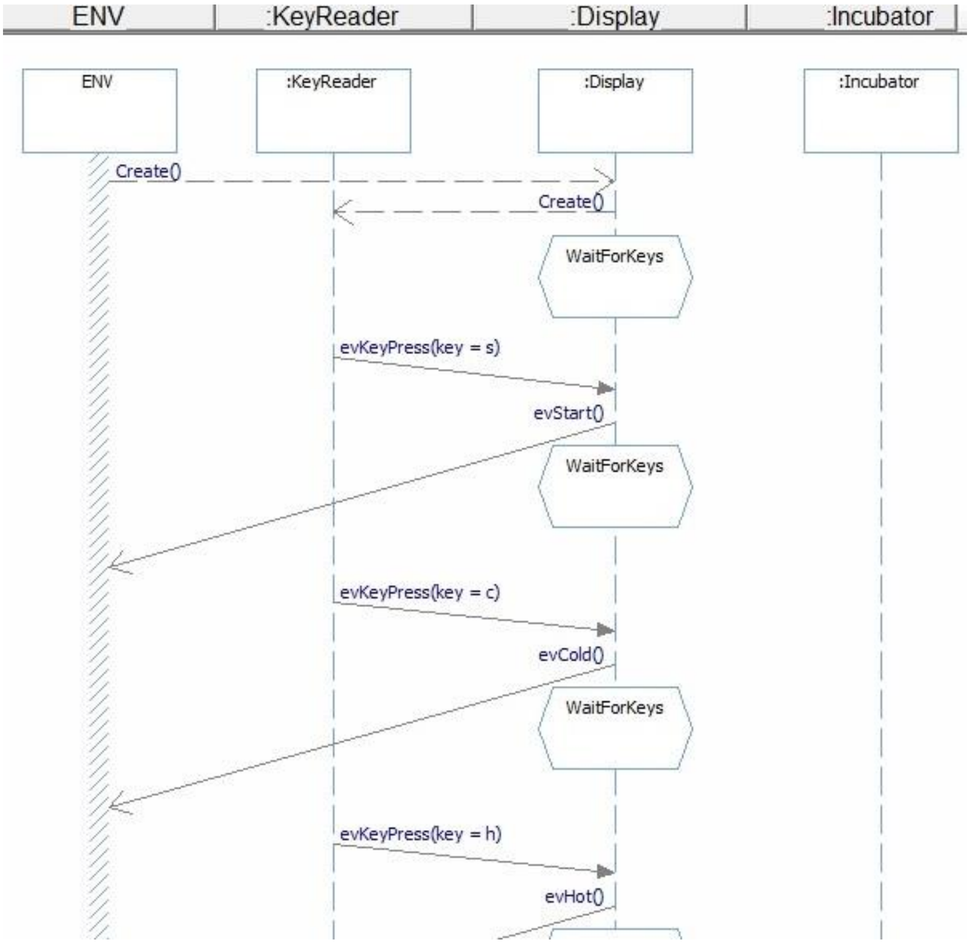
The state diagram:

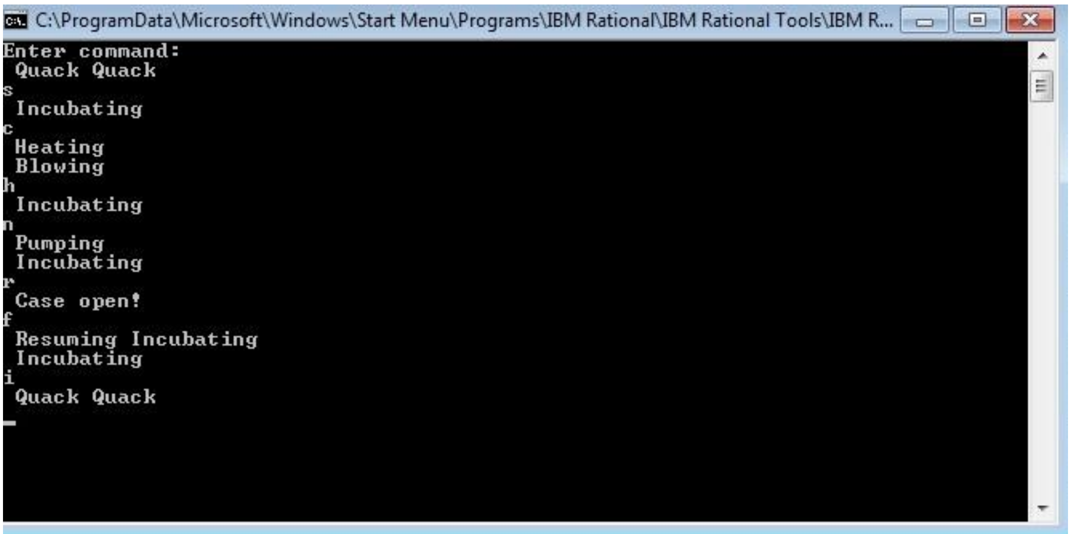
Diagram shows sequence of states that objects goes through, events that change its state and following actions.

This diagram shows the sequence of state the object takes, the events making the change of state and actions after change of state. We can distinguish base state "Running" with “Incubating” state in it, and multiple states for certain operations that take place after specific events.



Working project:




C:\ProgramData\Microsoft\Windows\Start Menu\Programs\IBM Rational\IBM Rational Tools\IBM R...

```
Enter command:
 Quack Quack
s
 Incubating
c
 Heating
 Blowing
h
 Incubating
n
 Pumping
 Incubating
r
 Case open!
f
 Resuming Incubating
 Incubating
i
 Quack Quack
_
```

Pictures above, show our project in work. We start with incubation, further sensors (simulated by the user) check temperature and humidity. In case of temperature too low, we start the heater and then the blower to blow warm air into incubator. Second sensor checks the humidity inside incubator, and if it drops too low, it starts with peristaltic pump in order to obtain correct humidity level.

If there are no disturbances we continue with incubating until it is stopped by the user.

Conclusions:

The UML helps in objected programming quite a lot. Despite the fact that it is mainly used for software models, its flexibility allows to use it in many different fields. It can be used for modeling business processes or organization scheme. In our project we used it to create system embedded in the device.