Team 2:
(Thomas Visser, Michel Peeters, Carl Megens, Marcel Verbunt, Fabian Koopman, Sander Kruitwagen)

Comments on the brief from team 1:

- In the Use Case Diagram:
    - The action 'Import animal' is given, however is not reflected in any of the other diagrams.
- In the Class Diagram:
    - Both the Parent and the Child classes don't contain any methods, so they actually aren't able to perform any actions (such as selecting, pinching, etc.).
    - Also the Animals are not able to scream.
    - We didn't completely understand the class 'Adopt'.
- In the State Chart Diagrams:
    - There is no final state, resulting in different errors, such as dead animal that run away, and adopted animals that get immediately refused again.
    - An animal that has given birth seems to become a (human) parent.
- In the Sequence Diagrams:
    - We couldn't completely understand the animal-animal interaction: Are the animals actors as well?
    - In one of the sequence diagrams it seems as if the animal is the one that decided it becomes a pet.
- In the Activity Diagrams:
    - In one of the activity diagrams it is described that for an animal to become a pet, three actions (playing, naming and feeding) should both start and end at the same time.

Weakness of the acting out team (team 3)

- They did not see that the shop was in the zoo
- They only used the state chart diagram for acting out; they did barely use the activity diagrams

Weakness of the specification team (team 2)

- Exchange without the owner knowing about this
- No final state in becoming happier; not able to become less happy because of fighting
- Pet provides pet list

Some details were wrong but overall the specification was worked out pretty well. The overall picture of the specification was shown clearly by the acting out of team 3.

Module overall reflection

We suggest for the next module to plan the acting out method earlier in the process to use it as inspiration for thinking of new interactions or for in between validation of the UML diagrams rather then as an end stage.
Furthermore we suggest to give some more exercises to master the UML language better. Lectures were OK but exercises can offer a real doing-it-yourself understanding. Lectures were long; suggest to do exercises in between!

Overall a proper module; good introduction into UML diagrams and acting out as a design (validation) method.