

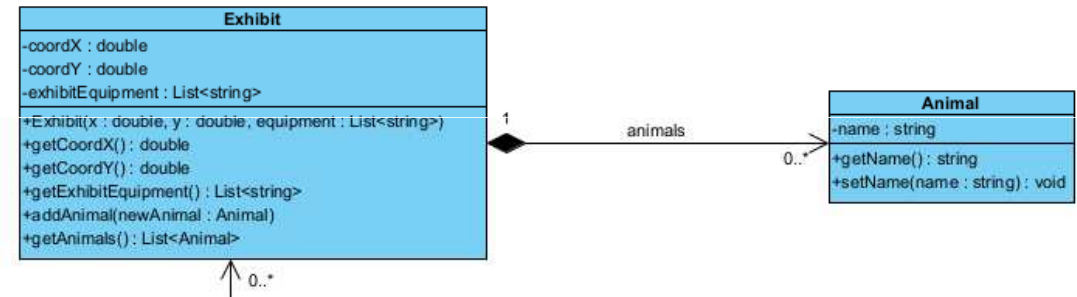
Interaction Diagrams, Sequence Diagram

PB007 Software Engineering I

Lukáš Daubner
daubner@mail.muni.cz

Previous diagrams

- Design class diagram
 - get / set methods for associations



- State Machine diagram
 - All used operations and attributes are present in the design class diagram
 - Transitions are not described with natural language
 - Call and change events only use operations and attributes from the respective class
 - Change events do not use attributes of different objects

Interaction Diagrams

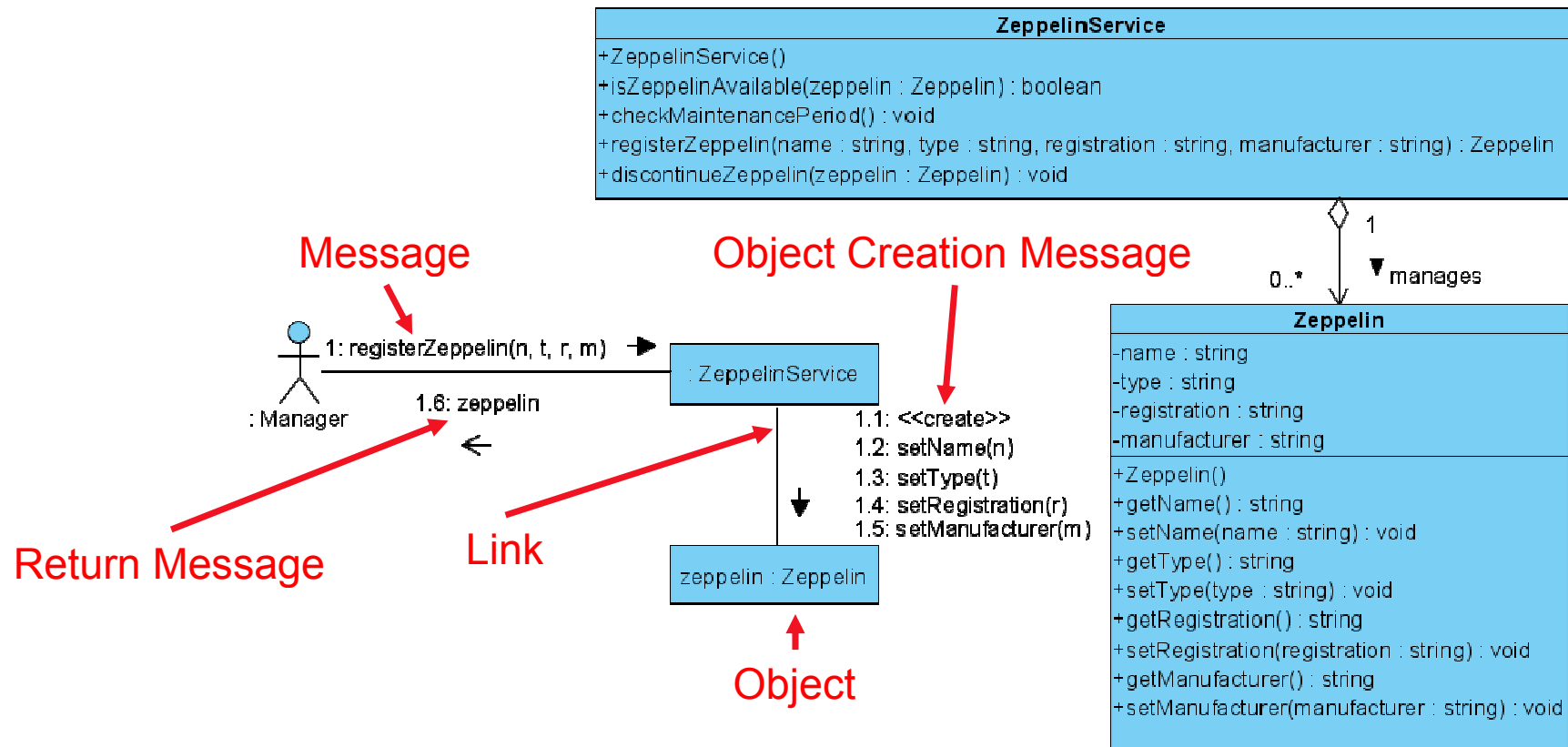
- Model collaboration between classes/objects/actors
 - i.e., communication between them
- Describes:
 - Method execution
 - Scenarios
 - Collaboration of subsystems
 - ...
- In general, they model behavior and communication between two and more objects
 - In **contrast to State Diagram**, which focuses only on inner behavior of one object



Communication Diagram

- Captures the communication with focus on **relationships** between objects
 - Interaction is considered as their entanglement
- From the object point of view, they describes detailed execution of a use case (or its part)
 - Again, everything must be consistent with Class Diagram
 - Link can exist only if there is a reference (association or dependency)
 - Messages are method calls on receiving object
 - We work with object – instances of classes

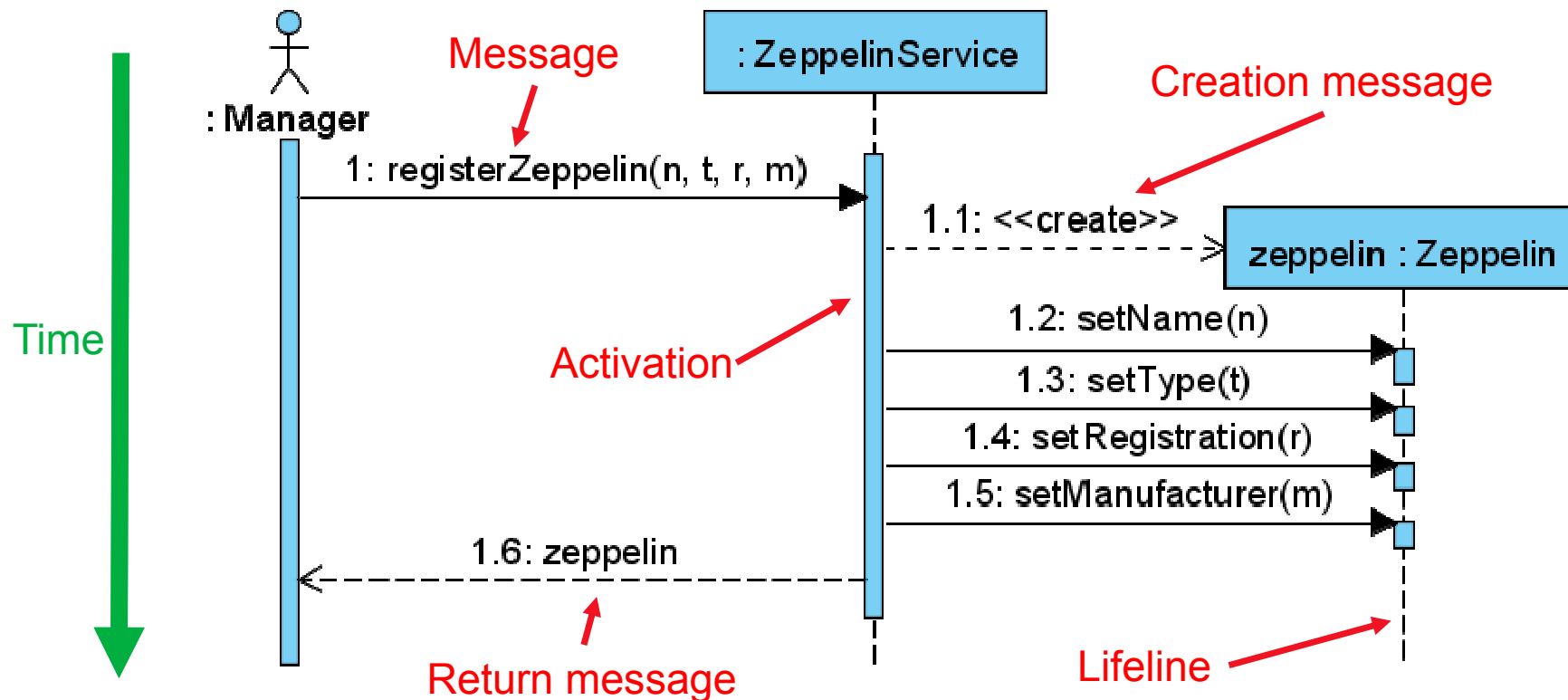
Communication Diagram – Example



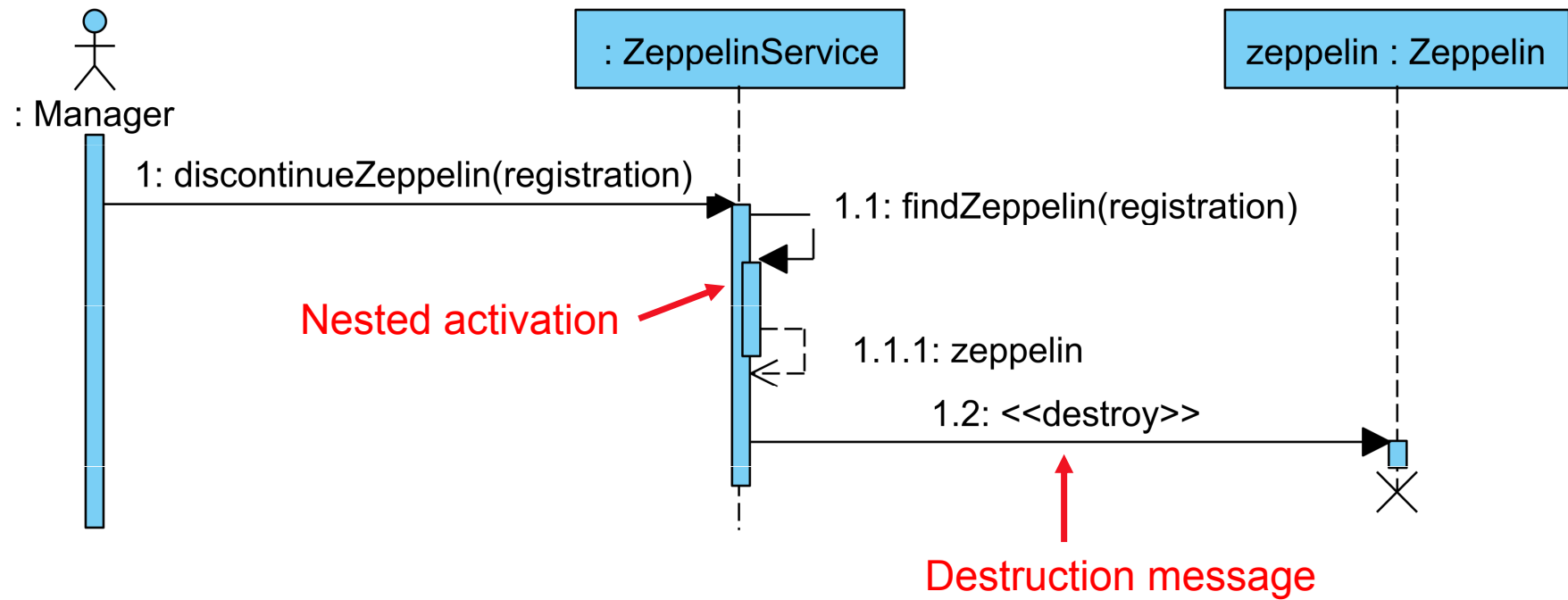
Sequence Diagram

- Captures the communication with focus on **time-based** ordering of messages
 - Interaction is considered as a sequence of messages
- From the object point of view, they describes detailed execution of a use case (or its part)
 - Again, everything must be consistent with Class Diagram
 - Link can exist only if there is a reference (association or dependency)
 - Messages are method calls on receiving object
 - We work with object – instances of classes
 - Mind the ordering, objects cannot receive message before their creation
 - Calls in one context must be in the same activation (focus of control)

Sequence Diagram – Example



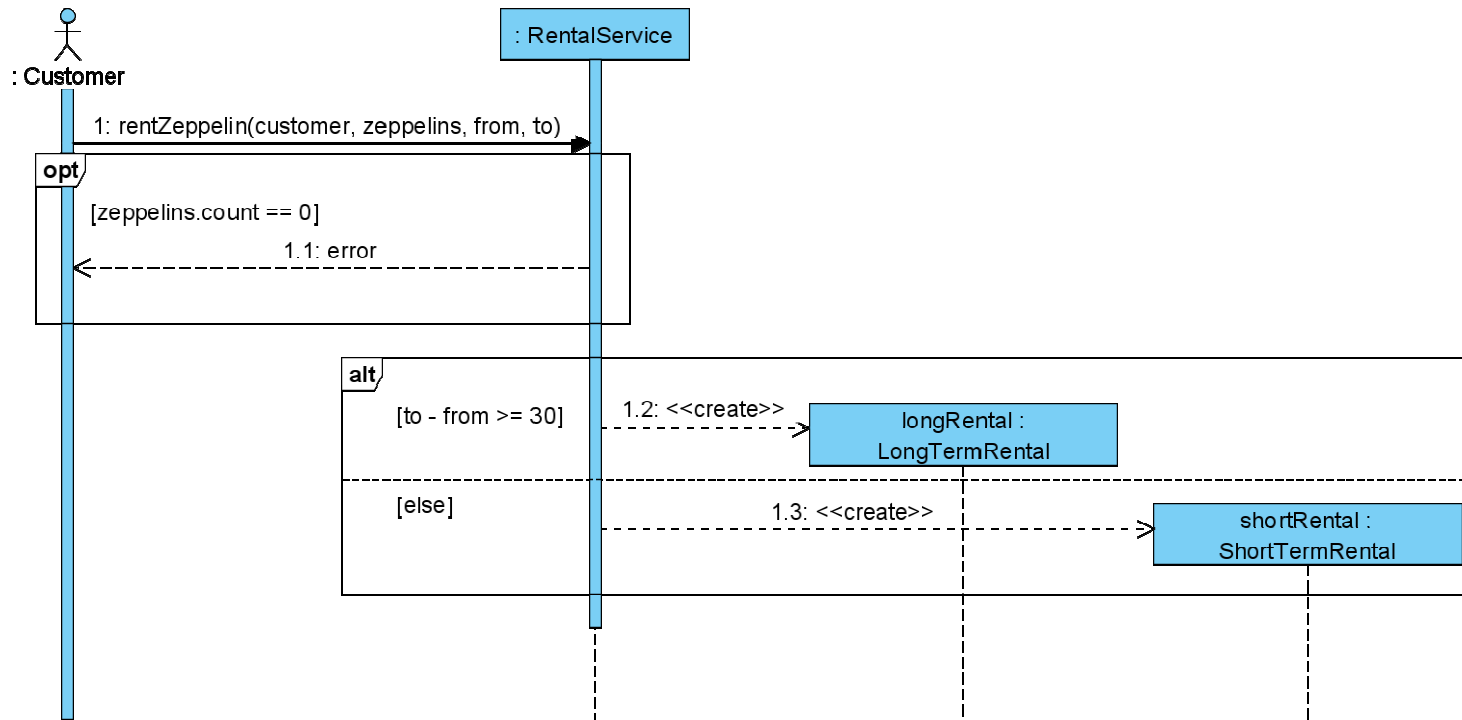
Sequence Diagram – Example



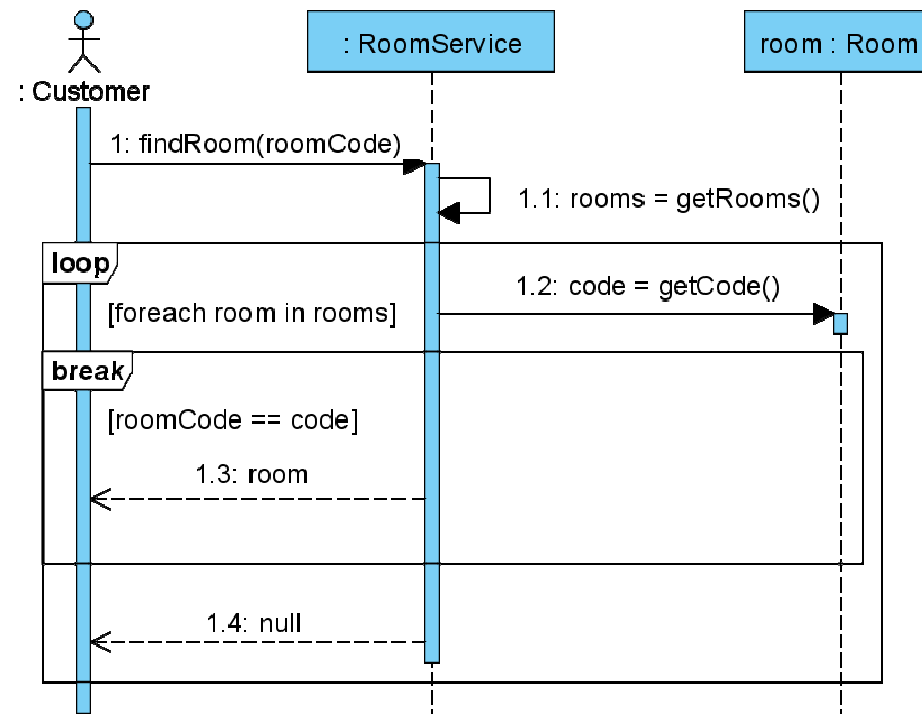
Combined Fragments

- Divides the sequence diagram into areas with different behavior
- They include *operator*, at least one *operand* and zero or more *conditions*
- Examples of major operators:
 - **opt (option)** – one operand, which is executed only if the specified condition is satisfied
 - **alt (alternatives)** – multiple operands, only the one with the satisfied condition is executed
 - **loop** – repeated execution of the operand
 - **break** – executed if its condition is satisfied, and it terminates the execution cycle

Combined Fragments – Example



Combined Fragments – Example



Task for this week

You gotta do what you gotta do

- Process the feedback
- Based on the Use Case Diagram and Design Class Diagram, think about the interaction between the objects that are participating in the use case
 - You can write it down as a pseudocode – it could help you
- **Choose three cases and model them into three Sequence Diagram**
 - If you need to change something in Design Class Diagram, just do it, no worries.