HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

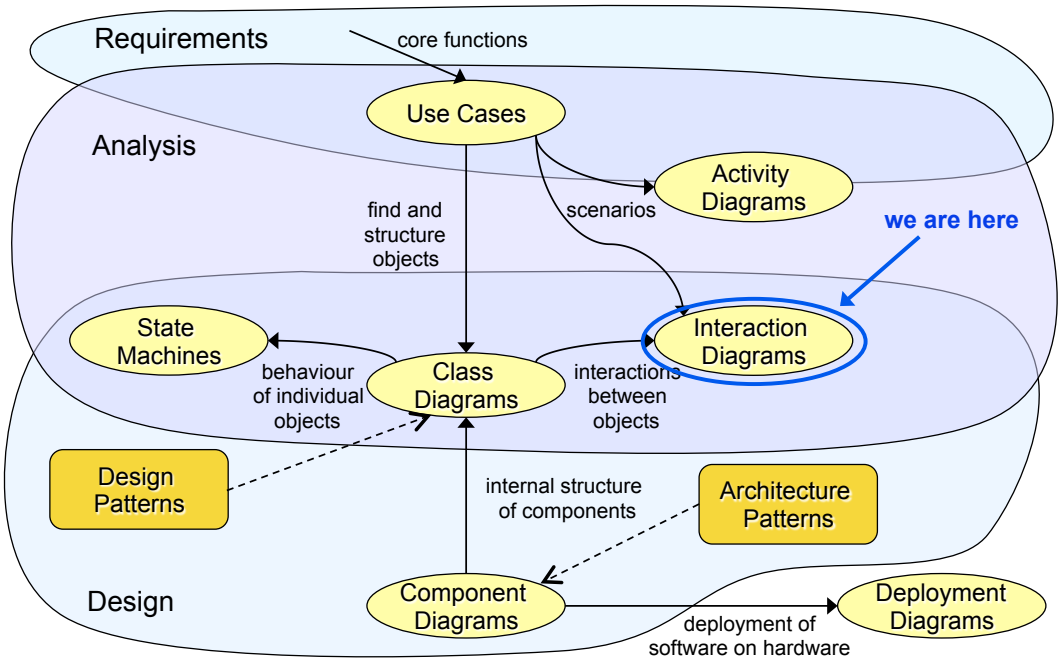
– Softwaretechnik – Analysis and Design with Interaction Diagrams

Hochschule Bremen
Internationaler Studiengang Medieninformatik
Prof. Dr. Thorsten Teschke
WS 14/15

Interplay between UML Diagram Types

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Interplay of most relevant UML diagram types:



Agenda

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- ➢ Introduction: Scenarios
- Sequence Diagrams
- Communication Diagrams

Use Cases and Scenarios

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Use case execution may have many variants depending on framing conditions (influence factors)
- Each use case is described in terms of collection of scenarios (cf. textual use case descriptions)
  - standard case
  - alternative executions

  } **Alternative: combined representation in activity diagram**
- Each scenario specification …
  - … describes a typical execution of the use case under particular conditions and
  - … therefore renders the use case more precisely

Use Cases and Scenarios: Two Categories of Scenarios

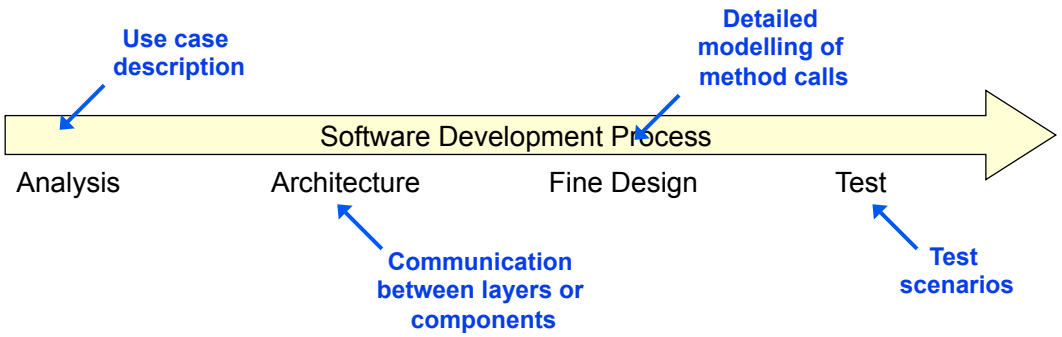
HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Two categories of scenarios can be distinguished:
  - scenarios that represent a successful processing of a use case
  - scenarios that lead to a failure
- Some scenarios for a use case "withdraw funds":
  - withdrawal of funds at ATM with sufficiently covered account
  - withdrawal of funds at ATM that exceeds balance …
    - … within limits of overdraft agreement ("Überziehungskredit")
    - … and overdraft agreement
    - … without overdraft agreement
  - withdrawal of funds at ATM, wrong PIN entered
  - withdrawal of funds at ATM, ATM out of money
  - …

Scenarios in Software Development

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Scenarios do not only occur in use case specifications, but also in later steps of software development



Scenario Modelling with UML

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- UML provides interaction diagrams for the specification of scenarios
- Interaction diagrams show (at least)
  - a set of objects and
  - messages that are exchanged between the objects within the modelled scenario

  } interaction of objects
- Interactions diagrams exist in the following variants:
  - sequence diagrams
  - communication diagrams (formerly known as collaboration diagrams)
  - timing diagrams (since UML 2.0, not covered in this lecture)

Agenda

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Introduction: Scenarios
- ➢ Sequence Diagrams
  - ➢ Characterisation
  - Message Exchange between Actors and Objects
  - Object Creation and Destruction
  - Control Structures
- Communication Diagrams

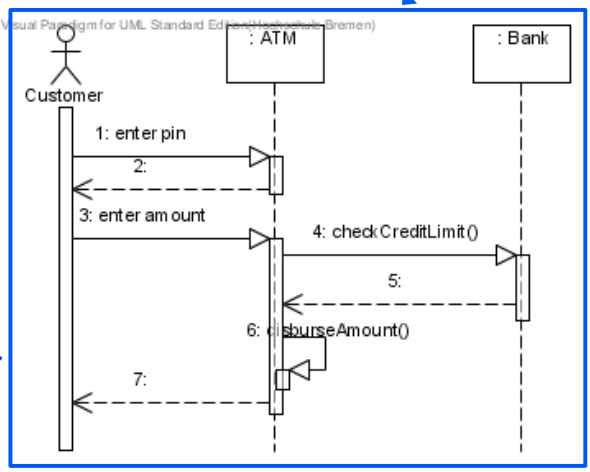
Interaction

- Definition (Interaction):
  "An interaction is a unit of behavior that focuses on the observable exchange of information between connectable elements." [UML 2.0 Spec.]

one interaction

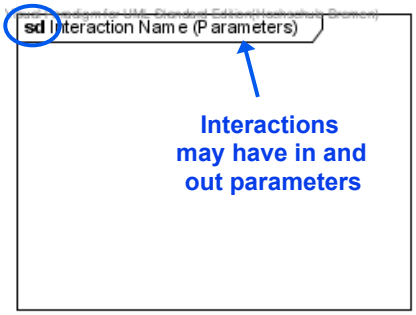
- Sequence diagrams show an interaction along two dimensions:
  - horizontal: communication partners (actors, objects)
  - vertical: (imaginary) time line

→ Emphasis on temporal order of message flow

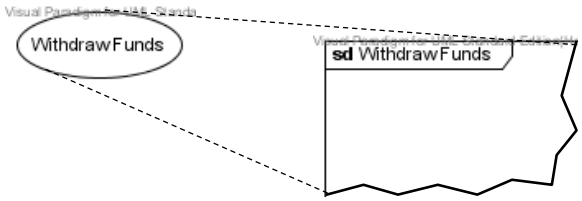


Interactions: Behaviour of Classifiers

- Interactions usually are contained in a solid-outlined rectangle
  - "sd" stands for sequence diagram
  - frame also used in other variants of interaction diagrams



- Interactions often are associated with a particular classifier which realizes the specified behaviour, e.g.
  - use cases
  - classes

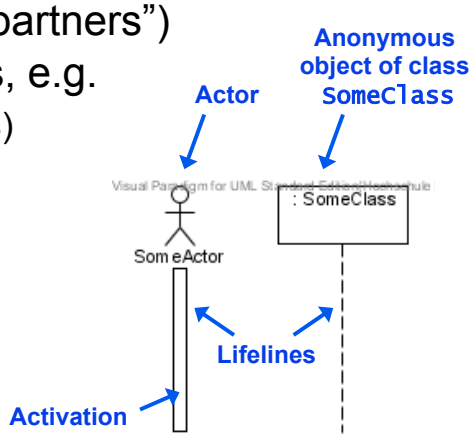


Agenda

- Introduction: Scenarios
- ➢ Sequence Diagrams
  - Characterisation
  - ➢ Message Exchange between Actors and Objects
  - Object Creation and Destruction
  - Control Structures
- Communication Diagrams

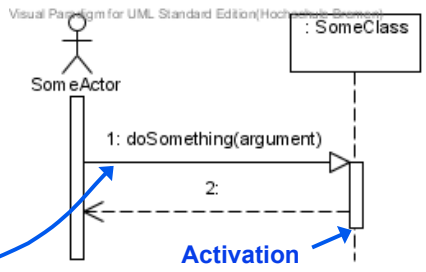
Communication Partners: Actors, Objects and Lifelines

- Focus in sequence diagrams is on temporal development of the communication partners' life cycles
- Lifelines represent individual participants in interaction
- Participants ("communication partners") may be instances of classifiers, e.g.
  - actors (as known from use cases)
  - objects



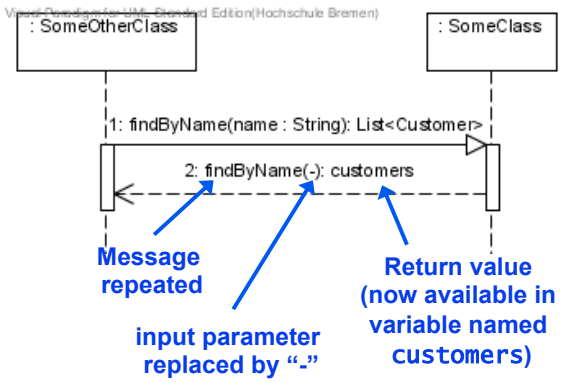
Message Exchange

- A message defines a particular communication between lifelines of an interaction
- The message specifies …
  - … the kind of communication, e.g.
    - invoking an operation,
    - creating or destroying an object, and
  - … the sender and the receiver (by a directed arc)
- Activation: actors are always in an activated state, objects only while they are processing a message



Message Exchange: Reply Messages

- Reply messages may be explicitly modelled as arrows with a dashed line and an open arrow head
- Returned values may be transported via reply messages:
  - message name is repeated
  - return value is specified after message introduced by colon
  - values of *in* parameters are repeated or replaced by "-"
  - [out values of *out / inout* parameters may be specified in argument list]



Message Exchange: Message Channels

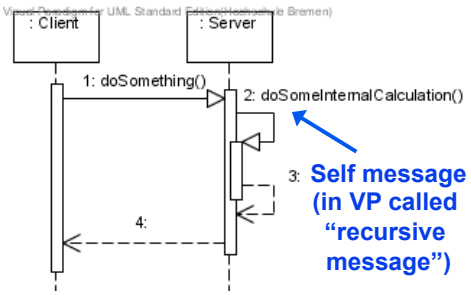
HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

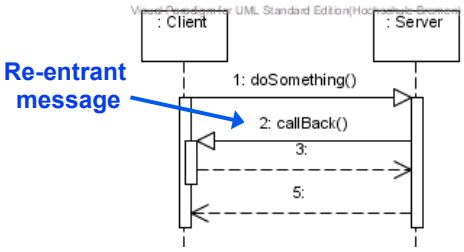
- Prerequisite for message exchange: acquaintance of communicating objects, i.e. sender must know receiver
- Relationships (assoc./aggreg./comp.) between objects represent "message channels" for communication
- Relationships may …
  - … exist statically (according to the class diagram),
  - … be created temporarily (object as an argument) or
  - … be internal (local object within a method)
- Note: this also holds for communication diagrams

Message Exchange: Overlapping Message Executions

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Sometimes, an object is asked to execute a message, while it is already processing another message
- Typical examples are:
  - self message: an object is sending a message to itself
  - re-entrant message: an object that is waiting for another object's reply is receiving a message from this object



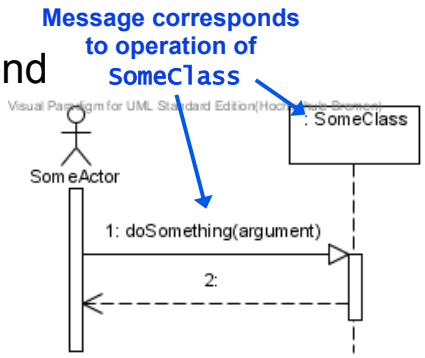


- Overlapping message executions result in additional activations (⇒ stack of invocations)

Message Exchange: Usage in Analysis and Design

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Messages in design *usually* correspond to operations of the receiving class
  - ⇒ promotion of consistency between class diagram(s) and interactions



- During analysis messages are often interpreted in a less formal way:
  - promotion of understandability: messages like “enter 4-digit pin” or “press checkout button” are admitted
  - informal messages are especially useful when modelling message exchange between human actors and user interfaces

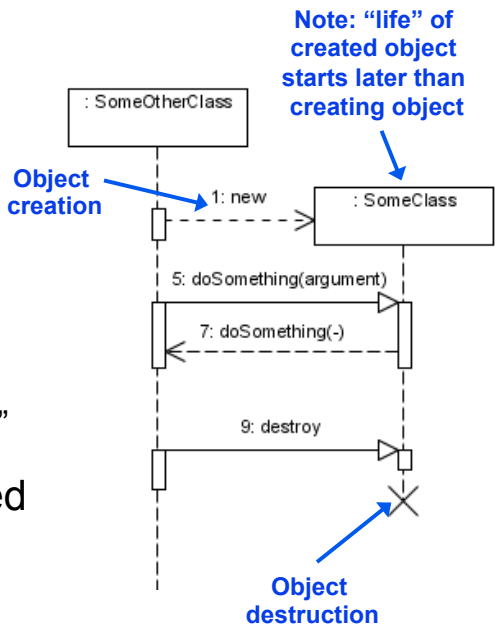
Agenda

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Introduction: Scenarios
- ➢ Sequence Diagrams
  - Characterisation
  - Message Exchange between Actors and Objects
  - ➢ Object Creation and Destruction
  - Control Structures
- Communication Diagrams

Object Creation and Destruction

- Creation messages create new communication partners, i.e. new instances of classes
  - arrow pointing directly towards head of new object's lifeline
  - creation messages not sufficiently specified in UML: use "new" as message name or "new *ClassName(parameters)*"
- Destruction of objects is signalled by destruction event
  - cross marks end of lifeline



Agenda

- Introduction: Scenarios
- ➢ Sequence Diagrams
  - Characterisation
  - Message Exchange between Actors and Objects
  - Object Creation and Destruction
  - ➢ Control Structures
- Communication Diagrams

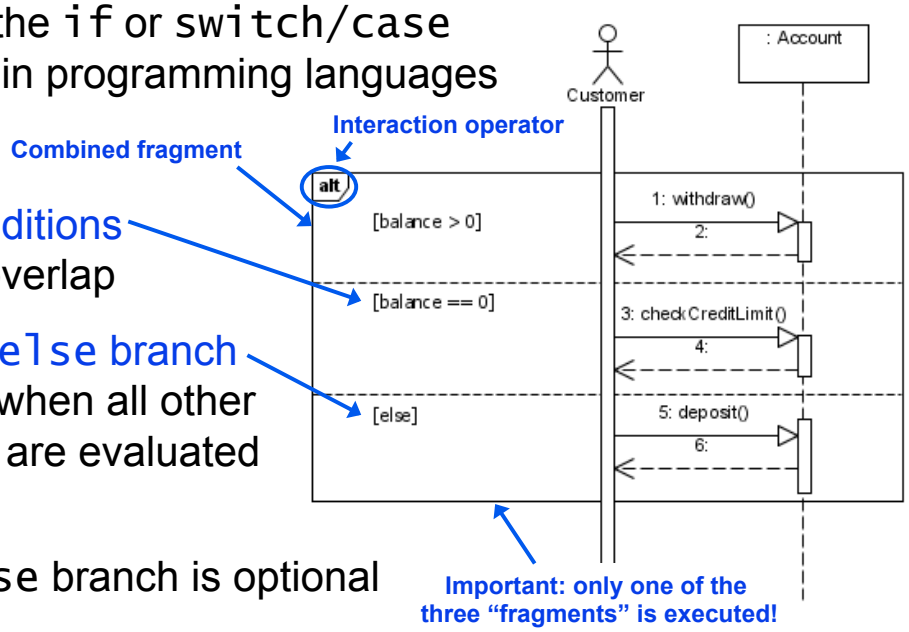
Combined Fragments

- Combined fragments are a means of specifying various control structures in a uniform way
- Idea: define expressions (interaction "fragments") by using interaction operators and interaction operands
  - interaction operators define selection, order, and frequency of execution of the participating interaction operands
- Most relevant interaction operators:

Name	Abbrev.	Semantics
Alternative	`alt`	Alternative interactions
Option	`opt`	Optional interaction
Loop	`loop`	Iteration of interaction

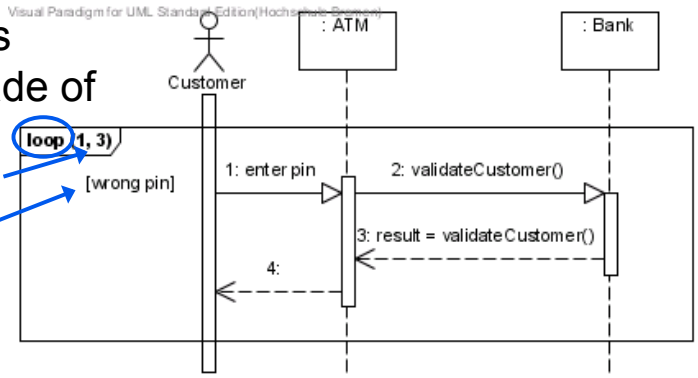
Combined Fragments: Alternative

- `alt` operator: specification of conditional behaviour
- Similar to the `if` or `switch/case` statement in programming languages
- Guard conditions must not overlap
- Reserved `else` branch is chosen when all other conditions are evaluated as "false"
- Use of `else` branch is optional



Combined Fragments: Loop

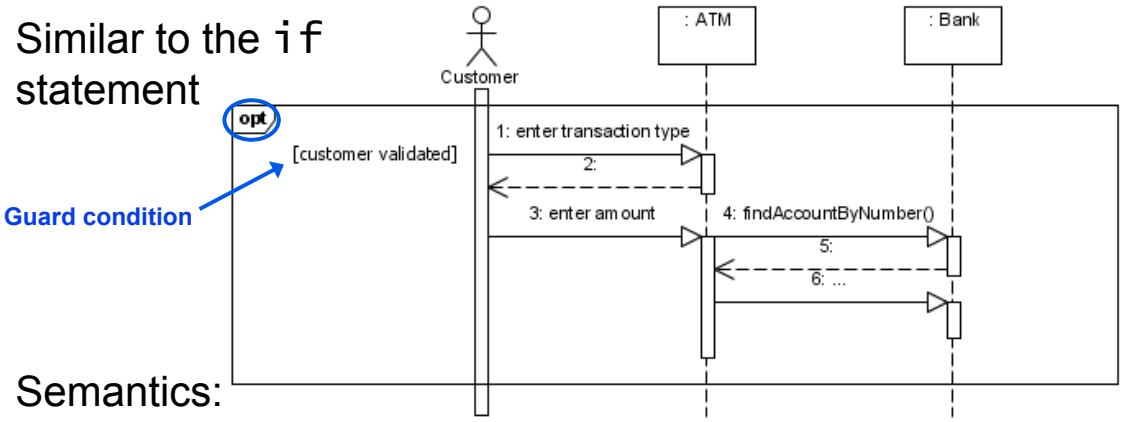
- `loop` operator: specification of iterative behaviour
- Number of iterations is specified by guard made of
  - lower and upper bound (`min`, `max`) *and*
  - boolean expression



- Semantics:
  - the loop will be executed at least the `min` number of times and at most the `max` number of times
  - the loop will terminate
    - if the boolean expression is false or
    - if the boolean expression is true and the maximum number of iterations has been executed

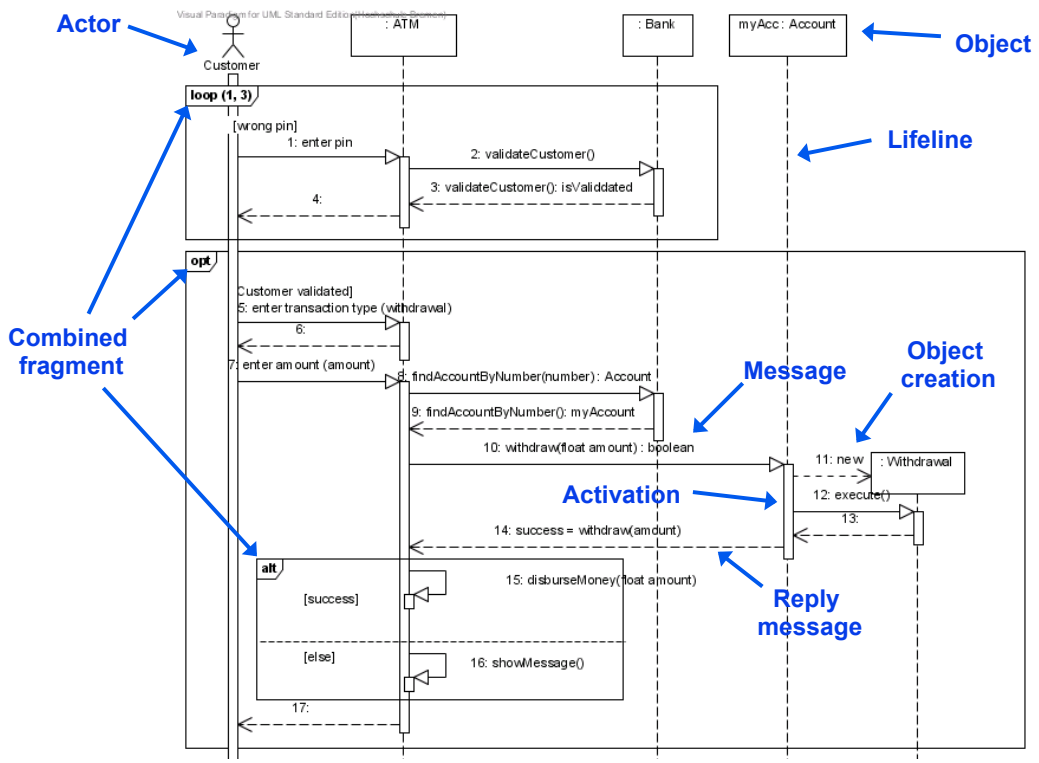
Combined Fragments: Option

- `opt` operator: specification of optional behaviour
- Similar to the `if` statement



- Semantics:
  - choice of behaviour where either the (sole) operand is executed (when guard condition holds) or nothing happens
  - semantically equivalent to an alternative combined fragment with empty second operand

Sequence Diagram: Overview



Exercise: Interactions for "eShop"

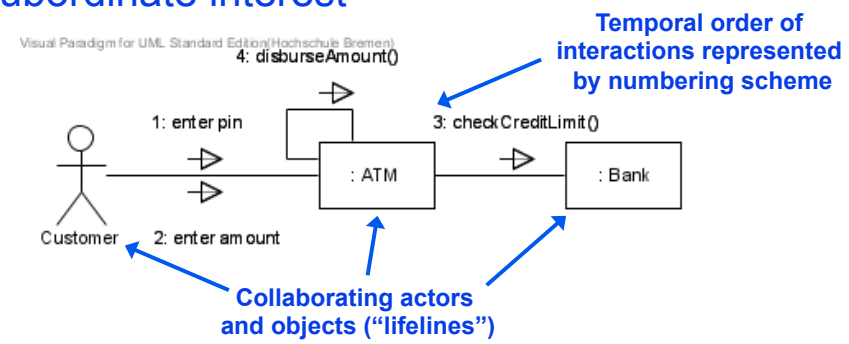
- The software design of the eShop has reached a stable state. In order to evaluate and improve the design the feasibility of various scenarios should be validated.
- Your next step in the design of "eShop" is the creation of a sequence diagram for one of the use cases
  - "Artikel in Warenkorb legen" (relatively easy)
  - "Warenkorb kaufen" (not that easy…)

Agenda

- Introduction: Scenarios
- Sequence Diagrams
- ➢ Communication Diagrams
  - ➢ Characterisation
  - Message Exchange between Actors and Objects
  - Control Structures

Characterisation (1)

- Communication diagrams were formerly known as collaboration diagrams
- Representation of interactions between "collaborating" actors and objects with focus on internal structure (associations between actors / objects)
- Temporal order of interactions is represented, but of subordinate interest



Characterisation (2)

- Communication diagrams share the same conceptual basis with sequence diagrams
- Reduced sequence diagrams:
  various syntactical elements known in sequence diagrams not available or only in simplified form
  - interaction uses
  - combined fragments
  - destruction of lifelines
  - …

⇒ Consequences:
  - conversion into sequence diagrams is possible
  - opposite is feasible only within limits

Agenda

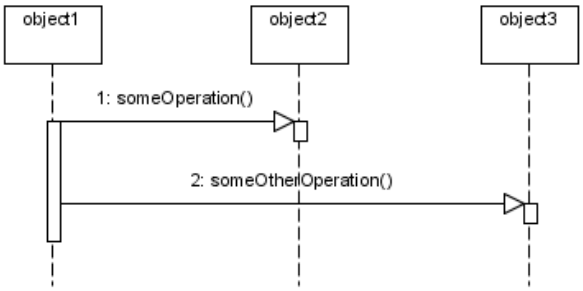
- Introduction: Scenarios
- Sequence Diagrams
- ➢ Communication Diagrams
  - Characterisation
  - ➢ Message Exchange between Actors and Objects
  - Control Structures

Messages

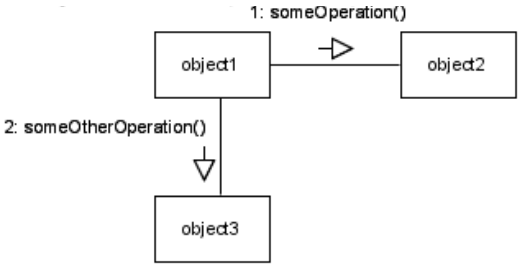
- Again: a message defines a particular communication between lifelines of an interaction
- Purpose: operation call and start of execution
- In communication diagrams:
  no explicit distinction between messages and reply messages
- Temporal ordering of messages is achieved by numbering scheme
  ⇒ reduced “visibility” of message sequence

Message Numbering Scheme (1)

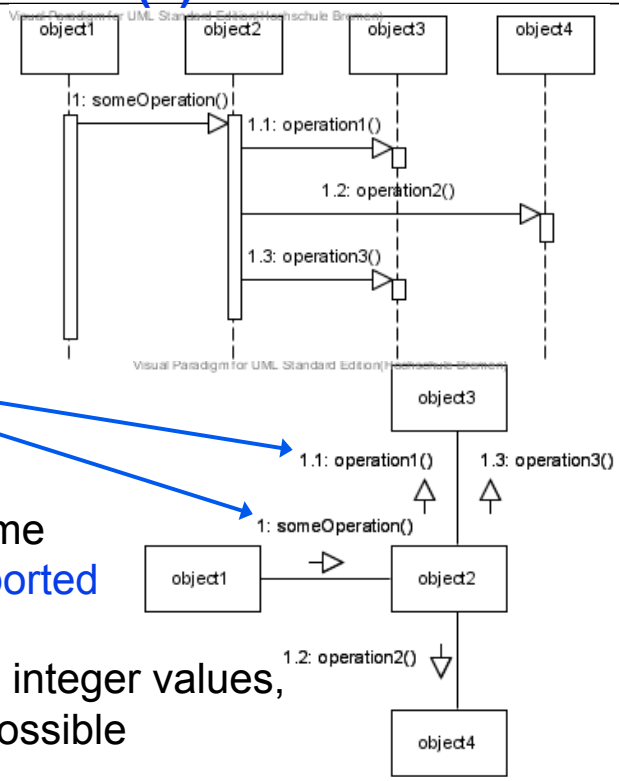
- Sequence expressions specify temporal ordering of messages
- Sequence expression is a dot-separated list of integers followed by a colon



- Messages that differ in one integer term are sequentially related at that level of nesting



Message Numbering Scheme (2)



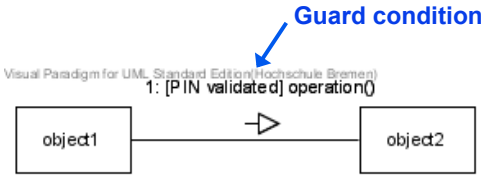
- Nested operation calls are numbered with number of surrounding operation call plus additional sequence number
- Nested numbering scheme is not automatically supported by Visual Paradigm: numbering by sequential integer values, but manual numbering possible

Agenda

- Introduction: Scenarios
- Sequence Diagrams
- ➢ Communication Diagrams
  - Characterisation
  - Message Exchange between Actors and Objects
  - ➢ Control Structures

Conditional Messages

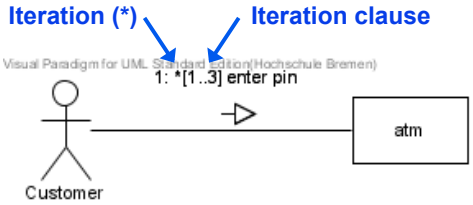
- Conditional messages are sent only if a guard condition is true
- Format of guard condition not prescribed by UML: condition is meant to be expressed in pseudocode or an actual programming language



- Remark: modelling of guard conditions not adequately supported by Visual Paradigm

Iteration of Messages

- An iteration represents a sequence of messages at the given nesting depth
- Number of iterations specified by iteration clause: clause is meant to be expressed in pseudo code or an actual programming language



- Remark: modelling of iterations not adequately supported by Visual Paradigm

When Should I Use What?

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Use sequence diagrams if …
  - presentation of temporal ordering of messages is important,
  - interactions are complicated and require control,
  - details of process flow need to be shown, and
  - the structural relationships between participants in the interaction is not of primary interest
- Use communication diagrams if …
  - a basic understanding is more important than details,
  - an interaction between parts within a complex structure needs to be presented in a simple way,
  - interactions are simple: no or only limited need for exact temporal relationships or control logic,
  - you want to show exactly one interaction and not variants

What You Should Know By Now

HOCHSCHULE BREMEN
UNIVERSITY OF APPLIED SCIENCES

- Scenarios in all phases of software development
- UML interaction diagrams:
  - sequence diagrams
  - communication diagrams
- Sequence Diagrams: interaction diagrams with special focus on temporal relationships of message exchange
- Communication Diagrams: simple variant of interaction diagrams with focus on structural relationships
- Next lecture: Design Patterns