	UML - Sequenzdiagramm		AnPr
	Name	Klasse	Datum

1 Allgemeines

Neben Aktivitätsdiagramm, Kollaborationsdiagramm, Zustandsdiagramm und Anwendungsfalldiagramm ist das Sequenzdiagramm eines von fünf Diagrammen in UML, welches dynamische Zusammenhänge von Systemen modelliert. Sequenzdiagramme fokussieren sich auf:

- einen bestimmten Prozessschritt („Szene“) in der gesamten Verarbeitung und die dabei benötigten Objekte bzw. Teilnehmer
- die Methoden und Signale, welche hierbei für die Kommunikation benötigt werden
- den zeitlichen Ablauf der Aufrufe bzw. des Nachrichtenaustauschs
- Instanziierung und Zerstörung der Objekte im Zeitablauf

Im Rahmen von UML zeigt das Sequenzdiagramm mitunter die genaueste Darstellung von Abläufen in Systemen oder Programmen. Dementsprechend sind die Diagramme durchaus umfangreich, bzw. müssen die dokumentierten Prozessteilschritte sehr klein „geschnitten“ werden.

Die Elemente des Sequenzdiagramms finden sich oftmals in anderen Diagrammtypen wieder:

- der beschriebene Prozess ist als Usecase Diagramm oder Aktivitätsdiagramm dokumentiert
- die involvierten Klassen (und deren Methoden) sind im Klassendiagramm hinterlegt

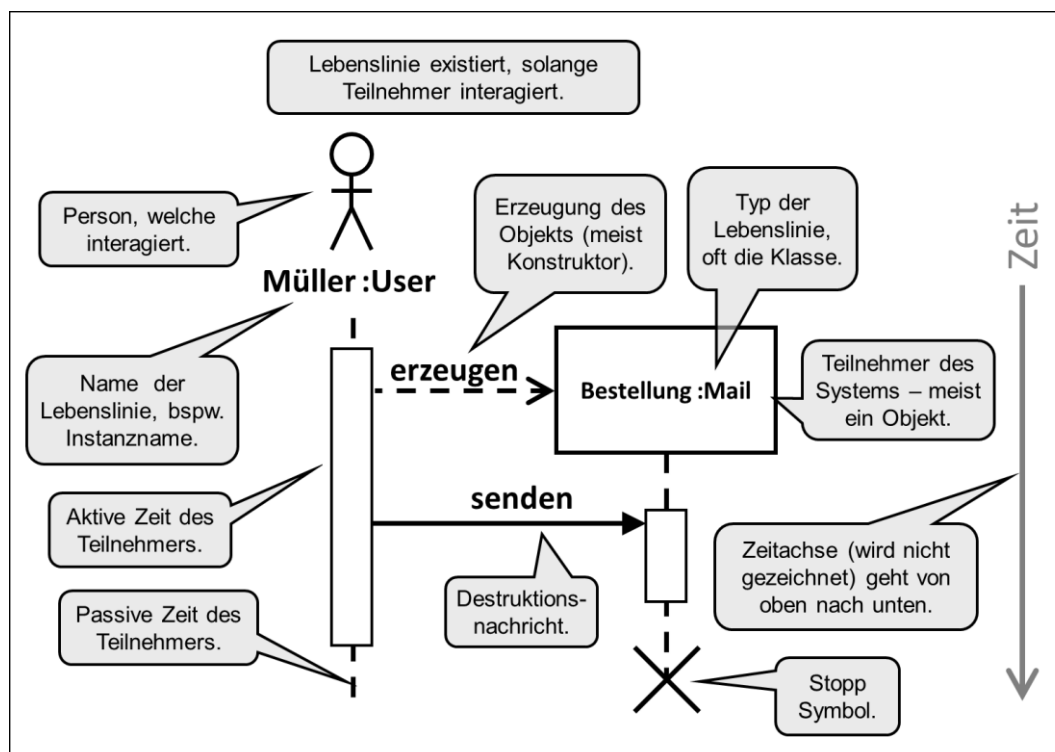
Die hier gemachten Angaben konzentrieren sich auf die wesentlichen Elemente. Die UML 2.5 Spezifikation gibt bei den Sequenzdiagrammen noch weitaus mehr Optionen vor.

2 UML-Sequenzdiagramm, Notation

Hier die wichtigsten Elemente des Sequenzdiagramms.

2.1 Lebenslinie

Die Lebenslinie symbolisiert die Zeit eines einzelnen Teilnehmers, welcher in der Kommunikation interagiert. Sobald seine Interaktionen abgeschlossen sind, bzw. das Objekt zerstört oder obsolet geworden ist (modelliert mit dem Stopp Symbol), endet die Lebenslinie.

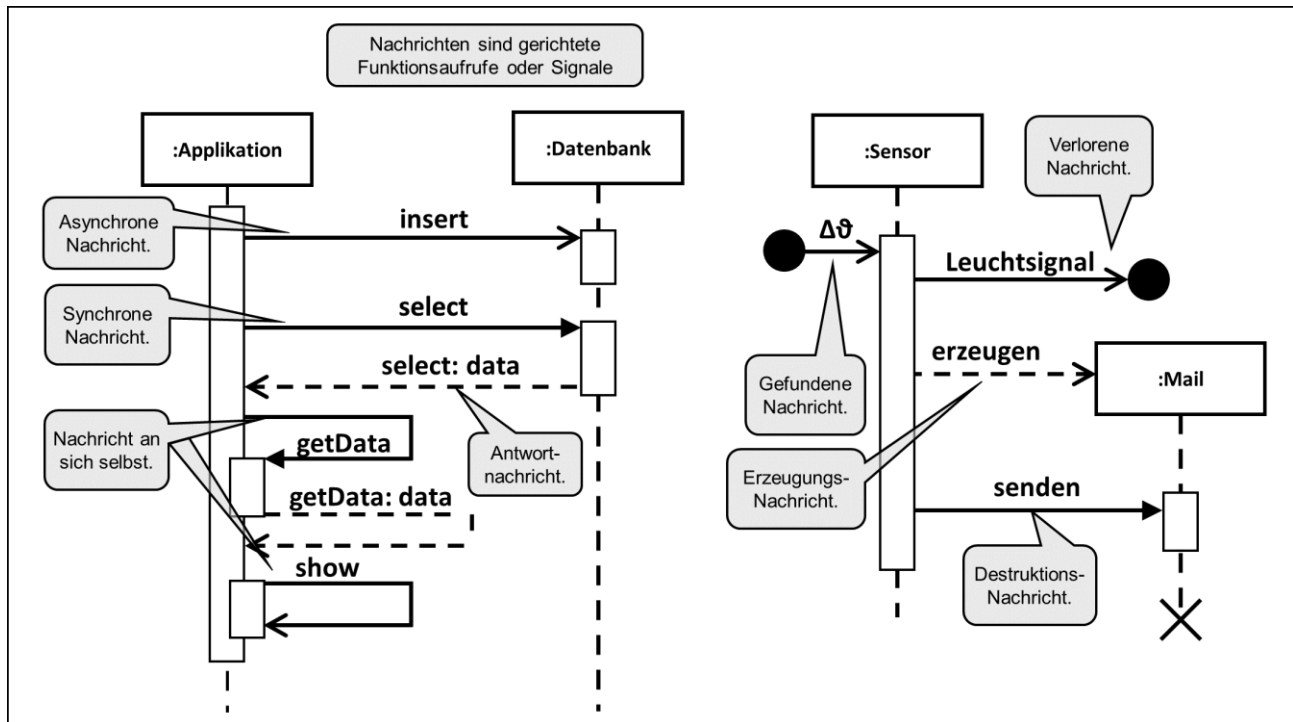


Lebenslinien können für Personen („Strichmännchen“) oder technischen Systemen bzw. Objekten stehen.

2.2 Nachrichten

Mit Nachrichten modelliert man den (gerichteten) Informationsaustausch zwischen zwei Teilnehmern. Hierbei handelt es sich um

- Funktionsaufrufe
- Signale



Bei **asynchronen Nachrichten** arbeitet der Sender sofort nach der Nachricht weiter. Er wartet nicht auf eine Rückmeldung des Empfängers. **Synchrone Nachrichten** führen dazu, dass der Sender erst weiteragiert, sobald der Empfänger eine Antwortnachricht geschickt hat. Synchrone Nachrichten können **keine Signale** sein. Die grafische Darstellung der drei verschiedenen Nachrichten ist wie folgt:

- Asynchrone Nachricht: durchgezogene Linie, offener Pfeil
- Synchrone Nachricht: durchgezogene Linie, ausgefüllter Pfeil
- Antwortnachricht: gestrichelte Linie, offener Pfeil. Trägt den Namen der synchronen Anfrage

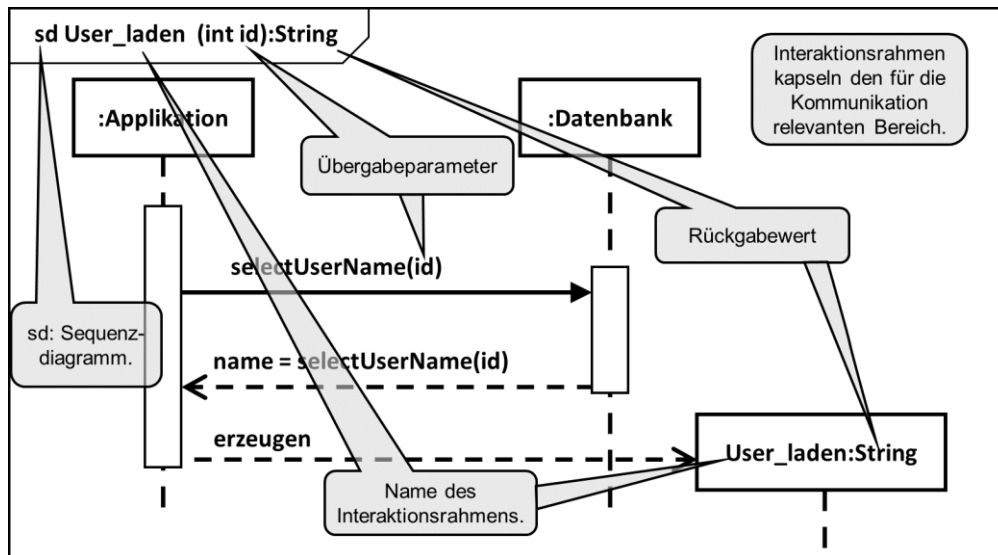
Nachrichten können auch Parameter und Antworten auch Rückgabewerte aufweisen. Weiterhin können Teilnehmer auch **Nachrichten an sich selbst** senden. Dies ist meist ein **interner Funktionsaufruf**, welcher modelliert werden soll. Wenn die innere Nachrichtenstruktur genauer dargestellt werden muss, so kann die Lebenslinie eines Teilnehmers auch durch Dekomposition in die einzelnen Subelemente aufgeteilt werden.

„Besondere“ Nachrichten sind:

- Erzeugungsnachricht: Gestrichelt, offener Pfeil zeigt auf den Kopf eines Teilnehmers. Meist ein Konstruktoraufruf
- Destruktionsnachricht: Durchgezogen, ausgefüllter Pfeil, führt nach der Verarbeitung zum Stoppsymbol. Meist ein Destruktoraufruf, bzw. die letztmalige Verarbeitung einer Objektreferenz
- Gefundene Nachricht: Durchgezogen, offener Pfeil, kommt von einem ausgefüllten Kreis. Sender der Nachricht ist für das Modell irrelevant, nur die Existenz der Nachricht ist wichtig.
- Verlorene Nachricht: durchgezogen, offener Pfeil, geht in einen ausgefüllten Kreis. Verarbeitung der Nachricht ist für das Modell irrelevant, nur das Senden der Nachricht ist wichtig.

2.3 Interaktionsrahmen

Um den relevanten Bereich des Kommunikationsprozesses zu kapseln, wird ein Interaktionsrahmen definiert. Dieser wird nach dem Kürzel „sd“ benannt und kann sowohl Übergabeparameter, als auch Rückgabewerte aufweisen.



Übergabeparameter werden benannt und können innerhalb des Diagramms mit dem entsprechenden **Namen** verwendet werden. **Rückgabewerte** werden über **ein eigenes Objekt**, welches den Namen des Interaktionsdiagramms trägt, zugewiesen.

Interaktionsrahmen können weiterhin noch lokale Attribute definieren, über sogenannte „Gates“ Nachrichten nach außen und innen übernehmen und innerhalb anderen, äußeren Interaktionsrahmen referenziert werden.

2.4 Fragmente

Fragmente oder auch „kombinierte Fragmente“ dienen zur Modellierung von komplexeren Strukturen. Folgende Fragmente sind definiert:

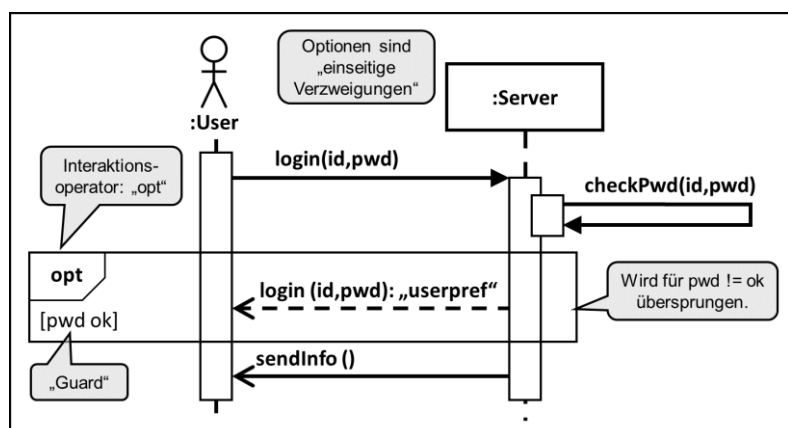
- Option
- Alternative
- Schleife
- Abbruch
- Negation
- Parallelität
- Schwache und strikte Sequenz
- Kritischer Bereich
- Relevante und irrelevante Nachrichten

Ich werde in diesem Dokument nur die wichtigsten behandeln, nämlich Option, Alternative und Schleife.

2.4.1 Fragment: Option

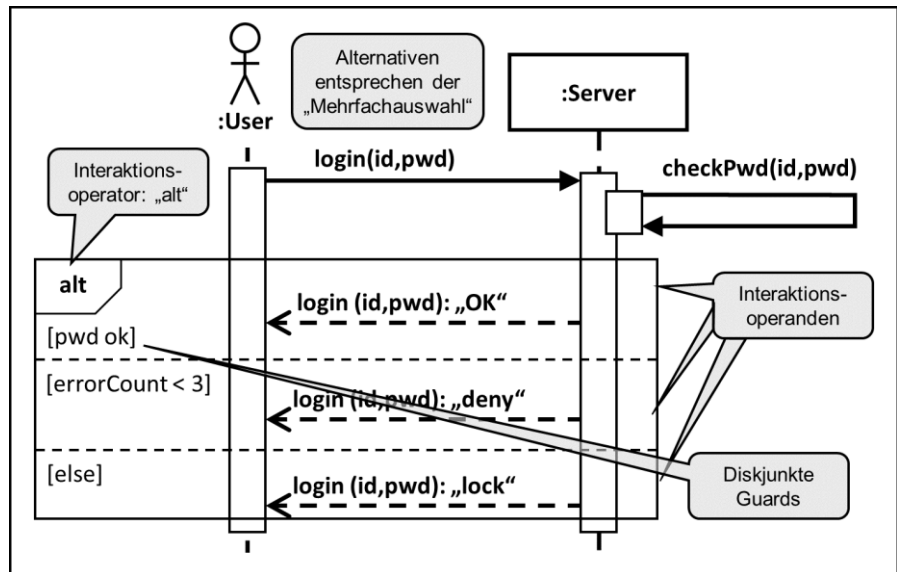
Optionen sind mit dem Struktogramm-element „einseitige Verzweigung“ vergleichbar. Der eingerahmte Bereich wird bei Nichterfüllung der Bedingung (genannt „Guard“) übersprungen. Es muss aus dem Kontext erkennbar sein, woher die Daten für die Bedingungsprüfung stammen.

Zur Kennzeichnung wird in den oberen linken Bereich „opt“ notiert.



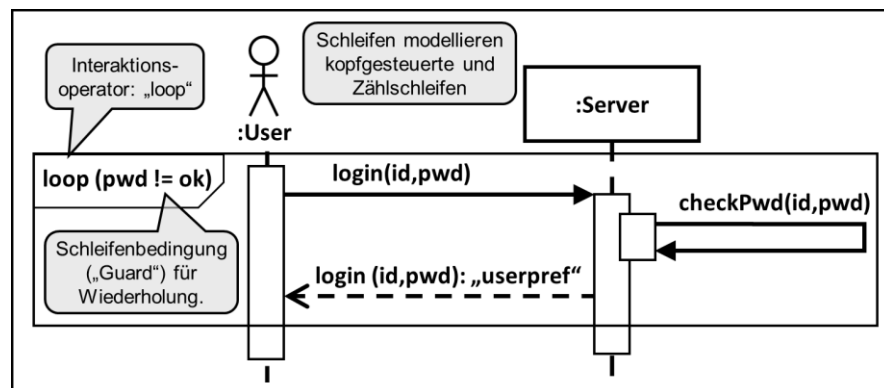
2.4.2 Fragment: Alternative

Alternativen kann man am Besten mit einer Mehrfachauswahl in der Struktogrammnotation vergleichen. Sie haben mindestens zwei Zweige, wobei für jeden Zweig eine eigene Bedingung („Guard“) Formuliert werden kann. Wichtig ist, dass diese Bedingungen sich nicht überschneiden („disjunkt sind“). Bspw. „errorCount \leq 3“ und „errorCount \geq 3“ wäre eine Verletzung dieser Forderung.



2.4.3 Fragment: Schleife

Struktogramme kennen drei verschiedene Schleifen – kopfgesteuerte, fußgesteuerte und Zählschleifen. Beim Sequenzdiagramm unterscheidet nicht zwischen kopfgesteuerten und fußgesteuerten Schleifen – diese muss aus dem Kontext gelesen werden – bzw. mit der Annahme „kopfgesteuert“ kommt man meist gut hin.

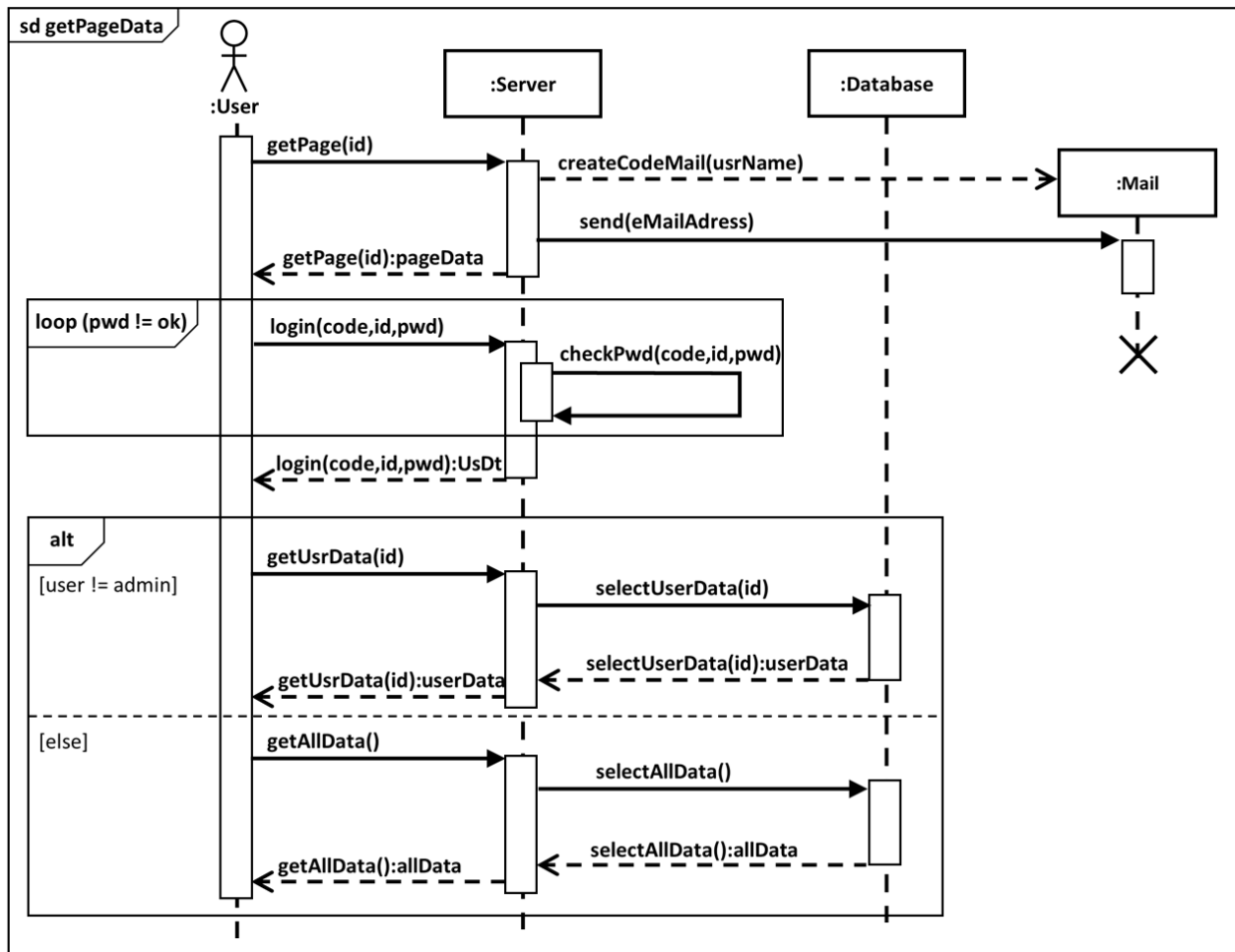


Die Bedingung („Guard“) kann wie folgt formuliert werden:

- Boolean Ausdruck als Weiterlaufbedingung der Wiederholung.
- Min bis Max als Start- und Endwert einer Zählschleife (i.d.R. mit Schrittweite ± 1)
- * für Endlosschleifen.

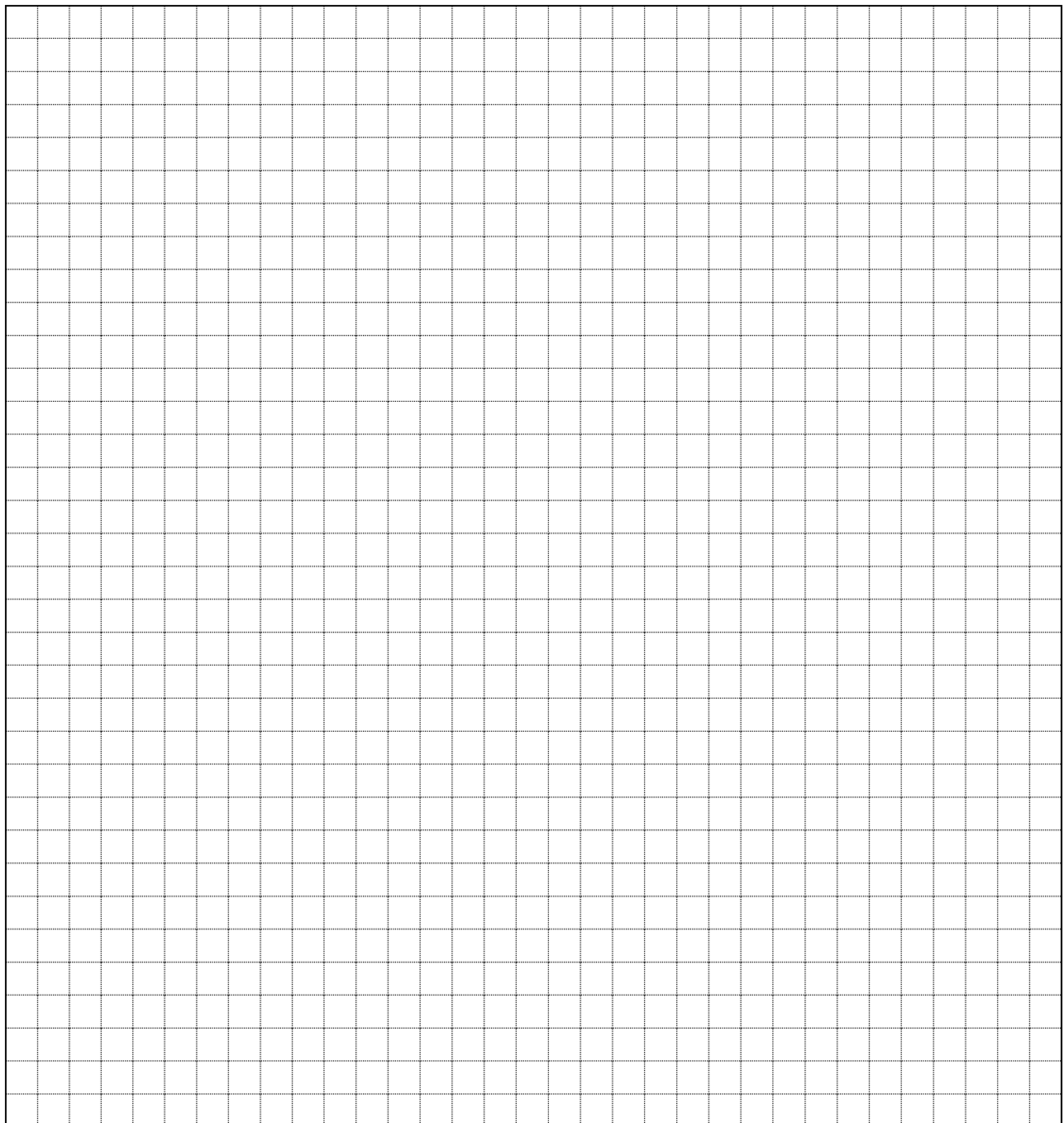
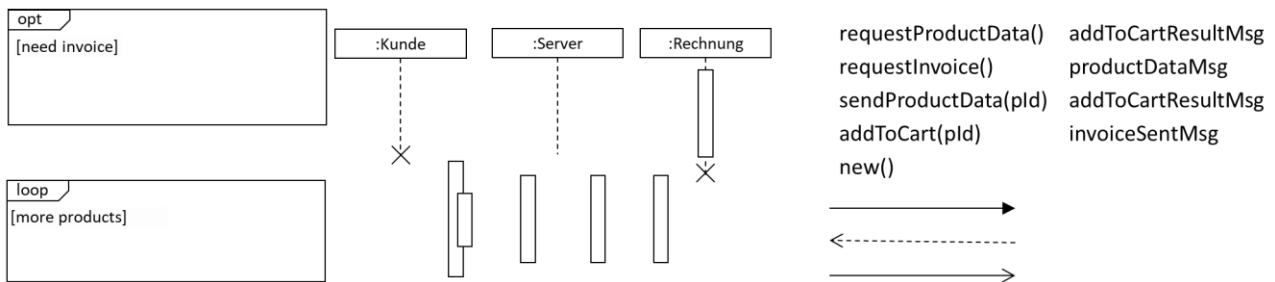
3 Beispiel

Im Folgenden Beispiel werden die wichtigsten Elemente dargestellt:



4 Aufgabenstellung

Für einen Webshop soll ein Sequenzdiagramm erstellt werden. Der User möchte die Produkte angezeigt bekommen, die gewünschten Produkte auswählen, damit sie in sein Shopping Cart hinzugefügt werden können. Wenn er keine Produkte mehr möchte wird er gefragt, ob er eine Rechnung haben will, welche ggf. erzeugt wird. Erzeugen Sie das Sequenzdiagramm basierend auf den gegebenen Elementen.



5 Lizenz



Diese(s) Werk bzw. Inhalt von Maik Aicher (www.codeconcert.de) steht unter einer Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 3.0 Unported Lizenz.