

Zusammenfassung

UNIVERSITÄT LEIPZIG

LPZ  **E-BUSINESS**
applied telematics

Objektorientierte Modellierungssprachen ermöglichen Vorteile, sie garantieren sie nicht.

Deshalb: objektorientierte Modellierung muss durch geeignete Methoden und Richtlinien ergänzt werden.

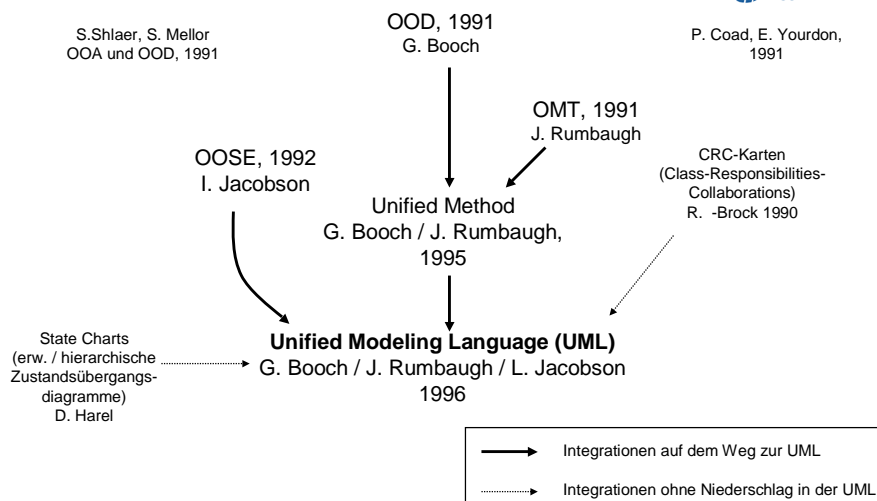
Versuch: UML und passende Methode

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 1

Unified Modeling Language

UNIVERSITÄT LEIPZIG

LPZ  **E-BUSINESS**
applied telematics



Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 2

Diagramme in der UML

Anwendungsfalldiagramme (Use Case Diagram)

- Akteure
- Anwendungsfälle
- Beziehungen

Klassendiagramme (Klassenstrukturdiagramm)

- Klassen
- Beziehungen zwischen Klassen

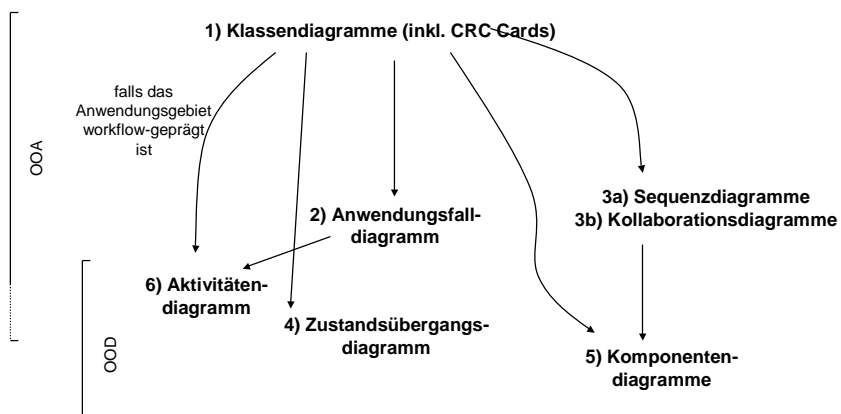
Implementierungsdiagramme

- Komponentendiagramm
 - Komponenten
 - Beziehungen zwischen Komponenten
- Einsatzdiagramme
 - Komponenten
 - Beziehungen
 - Knoten

Verhaltensdiagramme

- Aktivitätsdiagramme
 - Zustände
 - Zustandsübergänge
 - Ereignisse
 - Aktivitäten
 - Objektzustände
- Kollaborationsdiagramme
 - Objekte
 - Beziehungen inklusive Nachrichtenaustausch (räumlich geordnet)
- Sequenzdiagramme (Weg-Zeit-Diagramme)
 - Objekte
 - Beziehungen inklusive Nachrichtenaustausch (zeitlich geordnet)
- Zustandsdiagramme
 - Zustände
 - Beziehungen
 - Ereignisse

UML: Verwendung von Diagrammen



UML - Anwendungsfalldiagramme

UNIVERSITÄT LEIPZIG



Wichtig ist die Fokussierung auf Ziele des Anwenders und auf Interna des Systems, die mit diesen Zielen zusammenhängen. Beispiel:

- *Ziel: Was muss der Benutzer tun, um eine individuelle Dokumentenvorlage zu erstellen?*
- *Interna: definiere eine Vorlage, kopiere sie in das Verzeichnis für Vorlagen, wende diese Vorlage an.*

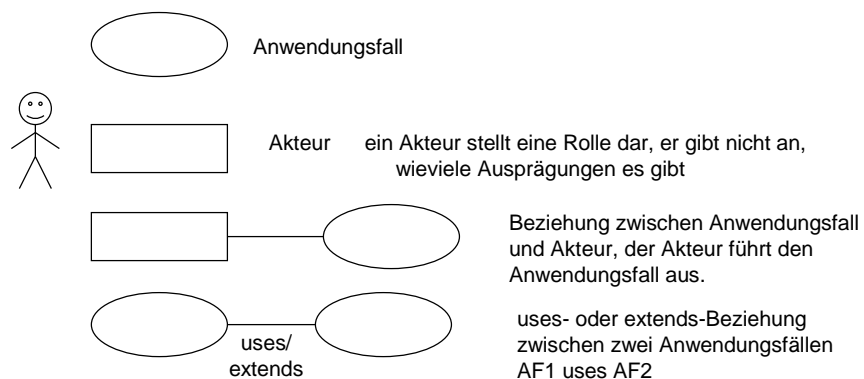
Zielfokussierte Anwendungsfälle eignen sich gut für die Abstimmung mit dem zukünftigen Anwender.

Auf das interne Verhalten ausgerichtete Anwendungsfälle eignen sich gut für Planungszwecke.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 5

Notation für Anwendungsfälle

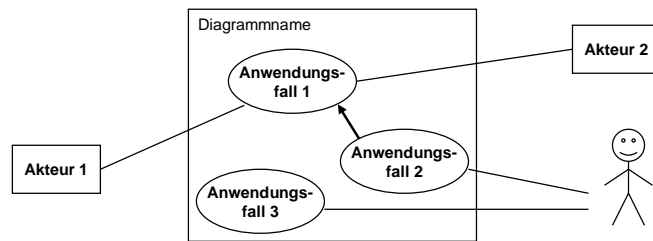
UNIVERSITÄT LEIPZIG



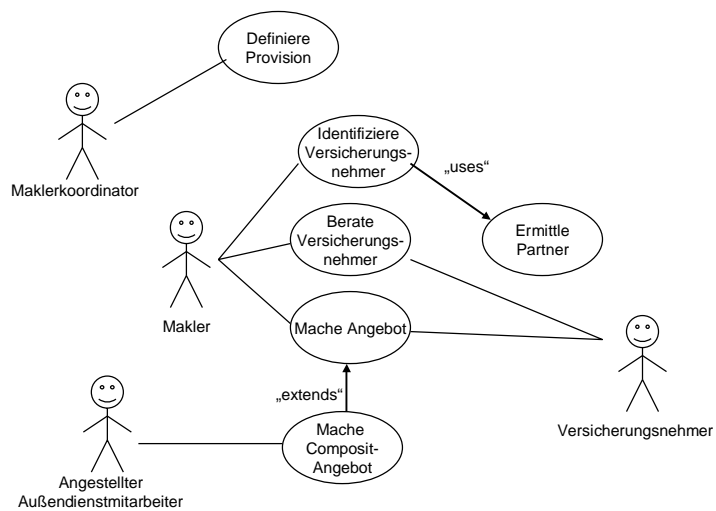
Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 6

UML - Anwendungsfalldiagramme

Notation für Anwendungsfälle



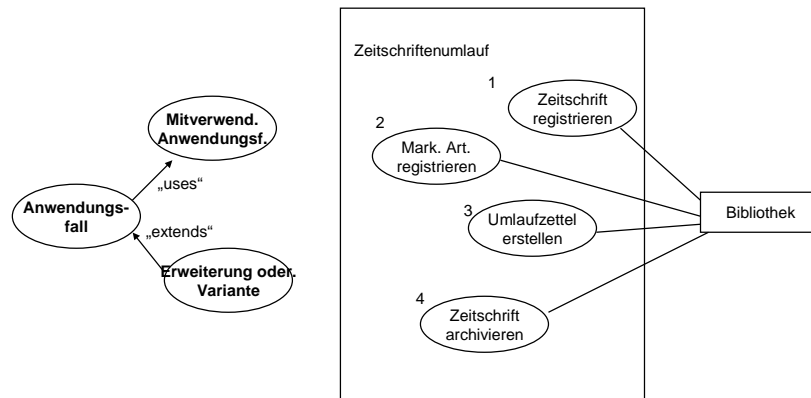
Anwendungsfalldiagramme - Beispiele



Anwendungsfalldiagramme - Beispiele

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics



Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 9

Strukturierte, textuelle Beschreibung eines Anwendungsfalles

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

Af.-Nr. Name des Anwendungsfalles

Vorbedingungen: ...
Nachbedingungen: ...
Nicht-funktionale Anforderungen: ...
Beschreibung: ..
Variationen: ...
Regeln: ...
Services: ...
Ansprechpartner: ...
Anmerkungen / offene Fragen: ...
Dialogbeispiele oder -muster: ...
Diagramme: ...

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 10

UML - Sequenzdiagramme (Weg-Zeit-Diagramme) und Kollaborationsdiagramme

- individuelle Objekte (in Sequenzdiagrammen und Kollaborationsdiagrammen)
- Beziehungen zwischen Objekten inklusive Nachrichtenaustausch (zeitlich geordnet) (in Sequenzdiagrammen)
- Beziehungen zwischen Objekten inklusive Nachrichtenaustausch (räumlich geordnet) (in Kollaborationsdiagrammen)

UML - Sequenzdiagramme (Weg-Zeit-Diagramme) und Kollaborationsdiagramme

Sequenzdiagramme und Kollaborationsdiagramme werden zusammenfassend als **Interaktionsdiagramme** bezeichnet.

Beide Arten von Diagrammen werden benutzt, um das Verhalten innerhalb eines Anwendungsfalls detailliert zu beschreiben. Sie liefern keine formale Beschreibung!

Sie beschreiben, wie Gruppen von Objekten miteinander interagieren.

Es werden Aussagen über die Anzahl der involvierten Objekte und über den Nachrichtenaustausch zwischen den Objekten gemacht.

UML - Sequenzdiagramme (Weg-Zeit-Diagramme)

Pro Objekt: eine vertikale Lebenslinie (wie lange existiert ein Objekt).

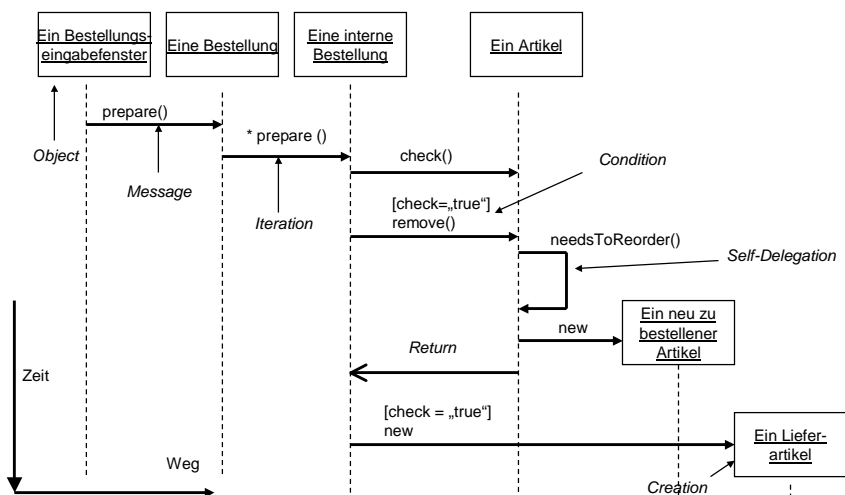
Nachrichten werden durch horizontale Pfeile zwischen den Lebenslinien der involvierten Objekte angezeigt (optional ergänzt um Parameter und zusätzliche Restriktionen). Zu den Restriktionen gehören:

- Vorbedingungen für das Versenden der Nachricht
- Iterationsangaben (wenn mehrere Objekte mit Nachrichten versorgt werden)

Ein Objekt kann sich selbst eine Nachricht schicken (Selbstdelegation). Dies wird graphisch dargestellt durch einen Pfeil, der als Quelle und Ziel die gleiche Lebenslinie hat.

Neben Nachrichten gibt es sogenannte *returns*. Ein return gibt an, ob und wann die Kontrolle an den Nachrichtenversender zurückkehrt (Pfeil mit offener Spitze).

UML - Sequenzdiagramme (Weg-Zeit-Diagramme)



UML - Sequenzdiagramme

Parallele Prozesse und Aktivierungen

Mit Hilfe von Aktivierungen kann ausgedrückt werden, wann die Operationen eines Objektes ausgeführt werden (wann also Prozesse zu diesen Methoden existieren).
 Aktivierungen werden durch Rechtecke auf den Lebenslinien der Objekte angegeben. Ihre Höhe deutet die Lebenszeit der jeweiligen Prozesse an.

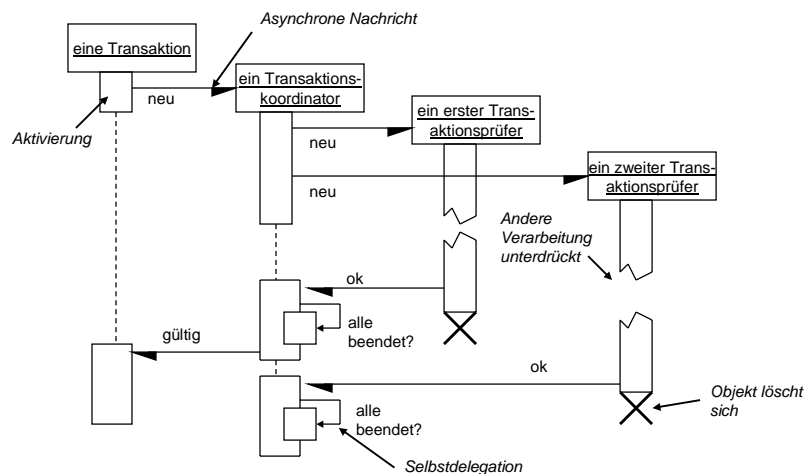
Selbstdelegation führt zu verschachtelten Aktivierungen.

Asynchrone Operationsaufrufe werden durch Pfeile mit halber Spitze dargestellt. Bei einem asynchronen Aufruf wartet der Aufrufer nicht auf ein Ergebnis. Typischerweise eingesetzt für:

- Erzeugen eines unabhängigen Kontrollbereichs.
- Erzeugen eines neuen Objektes.
- Kommunikation mit einem bereits existierenden Kontrollbereich.
- Pro Objekt: eine vertikale Lebenslinie (wie lange existiert ein Objekt).

Das Löschen eines Objektes wird durch ein X am Ende seiner Lebenslinie dargestellt.

Beispiel Transaktionsbehandlung - Regelfall



Beispiel Transaktionsbehandlung - Fehlerfall

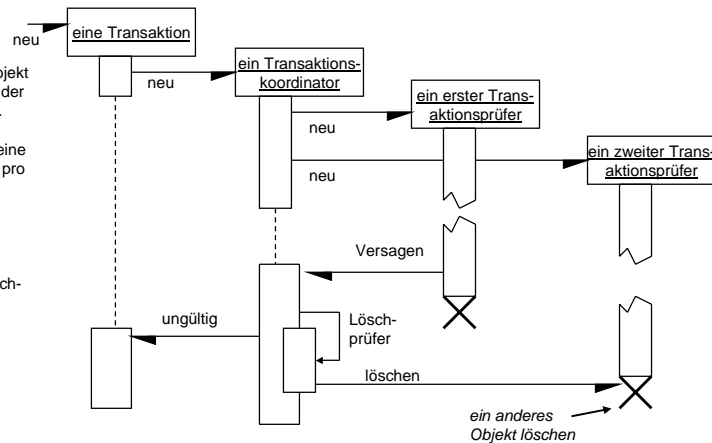
Wenn eine Transaktion erzeugt wird...

...dann erzeugt sie ein Objekt Transaktionskoordinator, der die Prüfungen überwacht.

Der Koordinator erzeugt eine Reihe von Prüfern, einen pro Prüfung. Diese Objekte arbeiten als getrennte Prozesse.

Wenn eine Prüfung nicht erfolgreich ist, dann vernichtet der Koordinator alle noch laufenden Prüfer.

...und teilt mit, dass die Transaktion ungültig ist.



Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 17

UML - Kollaborationsdiagramme

Kollaborationsdiagramme drücken den gleichen Sachverhalt aus wie Sequenzdiagramme.

Die Reihenfolge der Nachrichten ergibt sich aus der Nummerierung der Nachrichten.

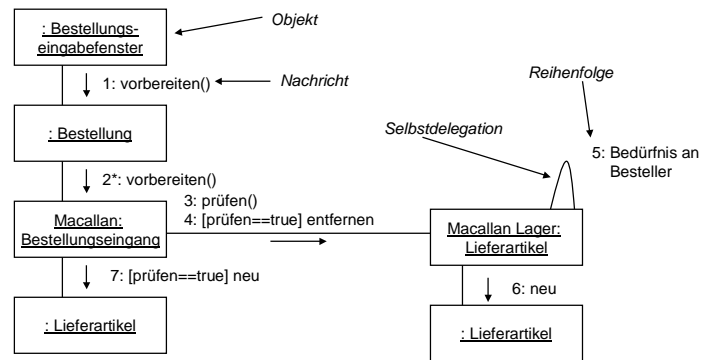
Die Anordnung der Objekte gibt die Möglichkeit, Zusammenhänge zwischen Objekten (zum Beispiel die Existenz von Beziehungen oder ihren räumlichen / örtlichen Zusammenhang) zu veranschaulichen.

Namensschema für Objekte: Objektname: Klassenname

In einem Namen darf entweder der Objektname oder der Doppelpunkt und der Klassenname ausgelassen werden.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 18

Kollaborationsdiagramme - Beispiel



UML - Zusammenfassung der bisherigen Diagramme

Klassendiagramm	Statik
Anwendungsfalldiagramm	Dynamik (Benutzersicht)
Interaktionsdiagramm	Dynamik (Kooperation von Objekten)
Sequenzdiagramm	
Kollaborationsdiagramm	

UML

Zusammenfassung der bisherigen Diagramme (2)

UNIVERSITÄT LEIPZIG



Und nun noch:

Zustandsübergangsdiagramme

Dynamik einzelner Objekte

einfache

parallele

Aktivitätsdiagramm

Abläufe / Anwendungsfälle

und ihre Kooperation

Komponentendiagramm

Systemstrukturierung

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 21

UML - Zustandsübergangsdiagramm

UNIVERSITÄT LEIPZIG



Zustandsübergangsdiagramme beschreiben die erlaubten Zustandsübergänge eines Objektes und die Ereignisse, die die Übergänge auslösen.

Zustände werden durch abgerundete Rechtecke dargestellt, Übergänge durch Pfeile.

Syntax für Pfeilanschriften: *Ereignis [Bedingung] Aktion* wobei alle drei Teile optional sind.

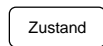
Wenn in einem Zustand interne Operationen ausgeführt werden sollen, so finden sich diese nach dem Schlüsselwort „*tue*“ im unteren Teil eines Zustandes.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 22

UML-Zustandsübergangsdiagramme (Notation)

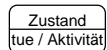
UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics



Ereignis [Bedingung] Aktion →

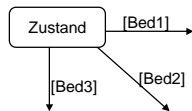
Zustandsübergang, der durch das Eintreten des Ereignisses ausgelöst wird (falls die Bedingung gilt!) und dabei die Aktion durchführt



Zustand, dessen Eintreten das Auslösen der Aktivität veranlasst.

[Bedingung] Aktion →

Ein Übergang ohne Ereignis tritt ein, sobald die Aktivität des Quellzustandes beendet ist.



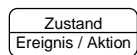
Die Bedingungen werden verwendet, um den gewünschten Zustandsübergang auszuwählen. Die Bedingungen an den Übergängen mit dem gleichen Quellzustand sollten sich wechselseitig ausschließen.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 23

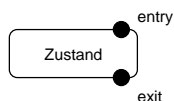
UML-Zustandsübergangsdiagramme (2) (Notation)

UNIVERSITÄT LEIPZIG

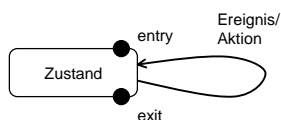
LPZ  E-BUSINESS
applied telematics



Der Zustand reagiert auf das Ereignis mit einer Aktion (die nicht zu einem neuen Zustand führt).



Ein Zustand kann mit einer Eingangsaktion (entry) und einer Ausgangsaktion (exit) assoziiert sein. (alternative Darstellung: rein textuell)



Ein Selbstübergang dieser Art führt zur Ausführung von

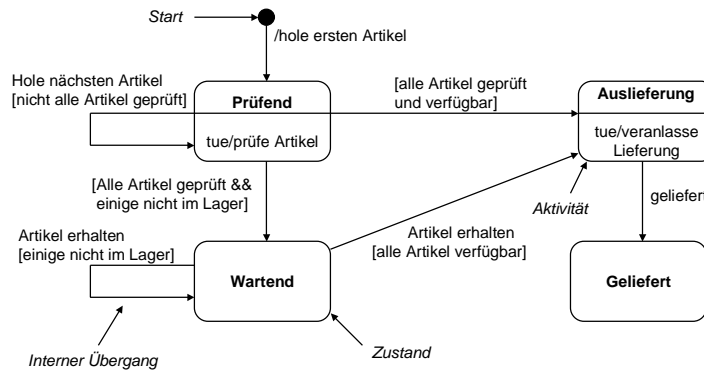
- exit
- Aktion
- entry

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 24

UML - Zustandsübergangsdiagramm (Beispiel Bestellung)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics



Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 25

UML - Zustandsübergangsdiagramm

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

Hinweis:

Wechselweiser Ausschluss der Bedingungen an den Übergängen, die vom Zustand „prüfend“ ausgehen:

- nicht alle Artikel geprüft => nächster Artikel wird geprüft
- alle Artikel geprüft und alle verfügbar => Lieferung veranlassen
- alle Artikel geprüft, nicht alle verfügbar => wartend

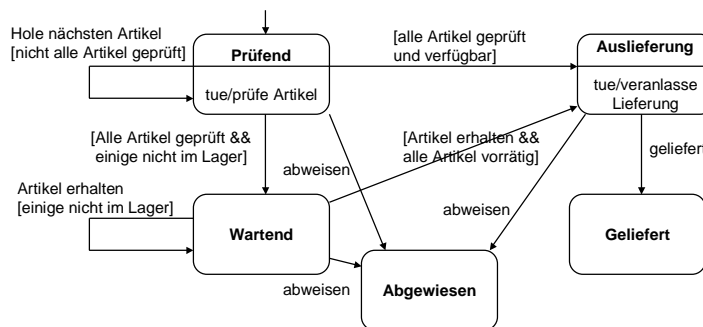
Zustände ohne Aktivität oder: Warten auf ein Ereignis

Bsp.: Im Zustand „wartend“ wird auf die Lieferung von Artikeln gewartet, sonst passiert nichts !

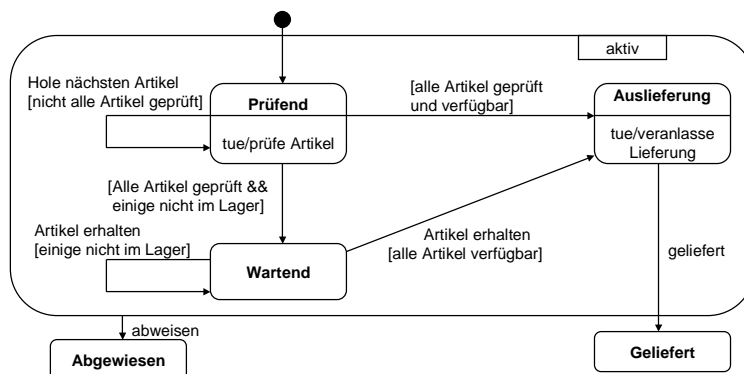
Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 26

UML - Zustandsübergangsdiagramm (Erweiterung um „Abweisen einer Bestellung“)

Anforderung: Abweisen soll jederzeit möglich sein
=> Von allen Zuständen werden Zustandsübergänge zu „Abgewiesen“ ergänzt.

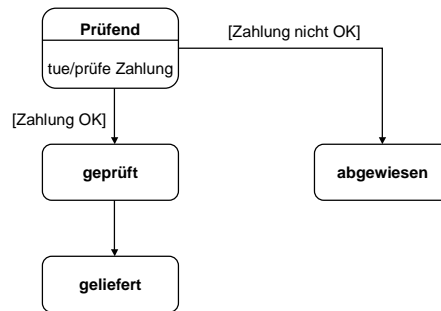


Alternative Strukturierung durch Oberzustände (Superstates)



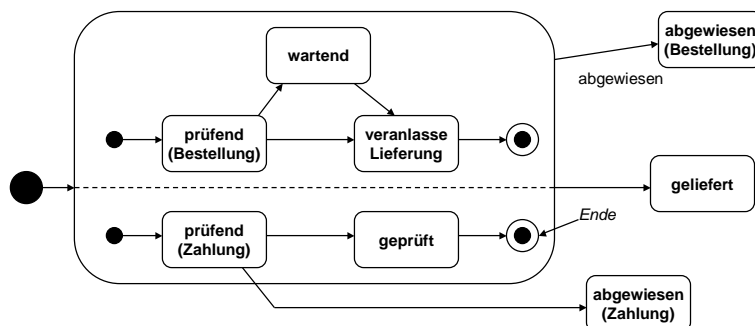
Hinweis: es bleibt auf dieser Ebene unspezifiziert, wann (von welchem Zustand kommend) der Zustand „abgewiesen“ eintritt!

Zustandsübergangsdiagramme (ein zweites Beispiel)



UML-Zustandsübergangsdiagramme

Oft werden externe Ereignisse als Auslöser verschiedener Zustandsübergangsdiagramme betrachtet. In zusammengehörigen Fällen (Bestellung, Zahlung) kommt man auf diese Weise zu parallelen Zustandsübergangsdiagrammen.



UML - Zustandsübergangsdiagramme

UNIVERSITÄT LEIPZIG



Parallele Zustandsübergangsdiagramme machen Sinn, wenn ein Objekt Mengen von unabhängigen Zuständen hat.

Komplizierte, parallele Zustandsübergangsdiagramme sind ein Indiz dafür, dass Objekte in mehrere Objekte aufgeteilt werden sollten (oben z.B. in Bestellung und Zahlung)

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 31

UML - Zustandsübergangsdiagramme (Bewertung)

UNIVERSITÄT LEIPZIG



Geeignet für die Beschreibung des Verhaltens eines Objektes über mehrere Anwendungsfälle hinweg

nicht geeignet für die Beschreibung der Interaktion zwischen Objekten

geeignet für die Beschreibung des Verhaltens der Objekte der **wesentlichen Klassen!** Vollständigkeitsfanatismus nützt nicht viel.

Geeignet für Benutzungsschnittstellen und Kontrollobjekte.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 32

UML - Aktivitätsdiagramme

UNIVERSITÄT LEIPZIG



Ein Aktivitätsdiagramm beschreibt die Reihenfolge und Abhängigkeiten von logisch zusammengehörenden Aktivitäten.

Eine Aktivität ist dabei ein einzelner Schritt in einem Verarbeitungsablauf.

Die Aktivitäten eines Aktivitätsdiagramms sind eindeutig Objekten zugeordnet (wichtiger Unterschied zu Datenflussdiagrammen !)

Aktivitäten und Aktivitätsdiagramme sind entweder

- einer Klasse
- einer Operation (besonders hilfreich für komplexe Operationen)
- einem Anwendungsfall zugeordnet. (besonders hilfreich bei Anwendungsfällen mit Parallelität)

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 33

UML - Aktivitätsdiagramme (Forts.)

UNIVERSITÄT LEIPZIG



Aktivitätsdiagramme gehen **nicht** auf eine der Ursprungsnotationen (Booch, Rumbaugh, Jacobson) zurück.

Aktivitätsdiagramme sind nützlich, wenn es um geschäftsprozessorientierte Softwaresysteme geht.

Die Ausführungsregeln für Aktivitätsdiagramme erinnern an die Ausführungsregeln von Petrinetzen.

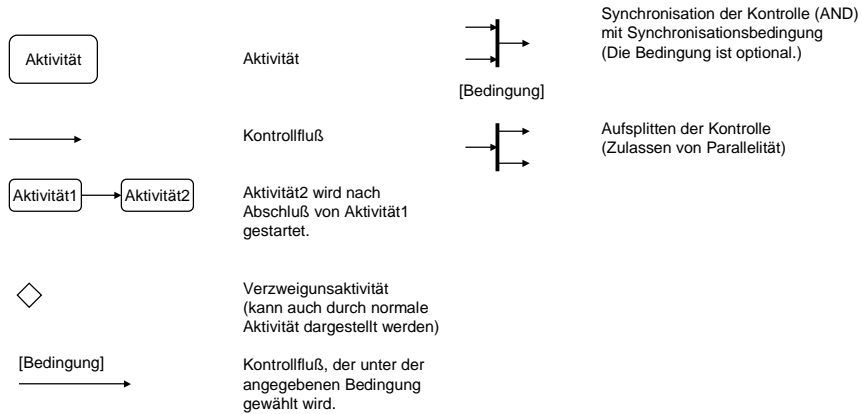
Zur Übung kann das Aktivitätsdiagramm „Getränke“ in ein Petrinetz überführt werden.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 34

UML - Aktivitätsdiagramm (Notation)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

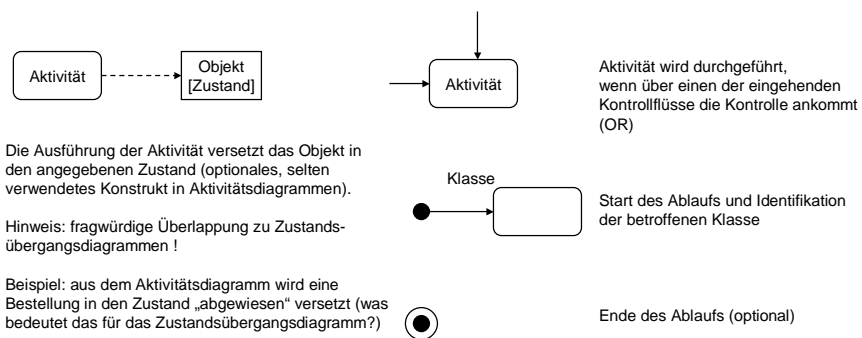


Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 35

UML-Aktivitätsdiagramm (Forts.) (Notation)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

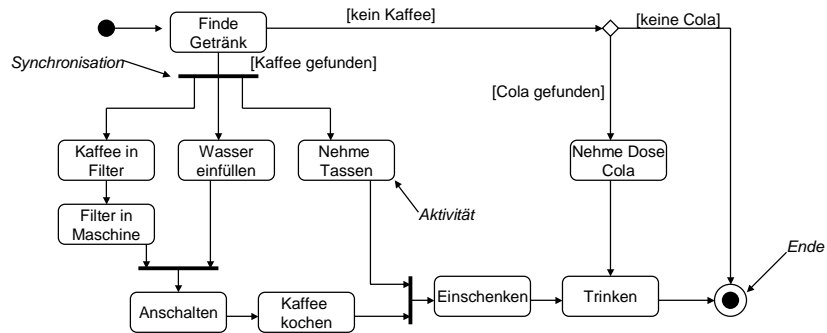


Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 36

UML - Aktivitätsdiagramm (Beispiel zur Spezifikation einer Operation)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics



Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 37

UML - Aktivitätsdiagramm (Beispiel zur Spezifikation eines Anwendungsfalls)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

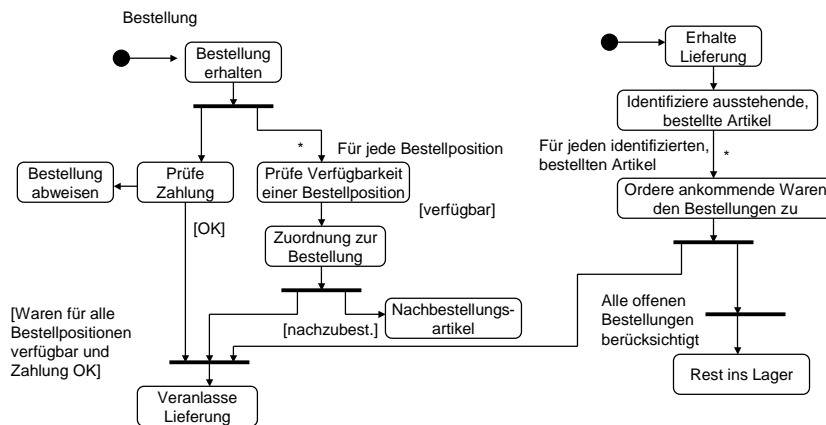
Informale Beschreibung:

Wenn eine Bestellung eintrifft, überprüfen wir jede Bestellposition, um zu sehen, ob der Lagerbestand reicht. Falls das so ist, werden die Waren der Bestellung zugeordnet. Wenn durch diese Zuordnung der minimale Lagerbestand unterschritten wird, dann wird eine Nachbestellung veranlasst. Währenddessen wird die Zahlung überprüft. Wenn die Zahlung OK ist und genügend Waren vorhanden sind, wird geliefert. Wenn wir für die Lieferung auf nachzubestellende Waren warten müssen, bleibt die Bestellung offen. Wenn die Zahlung nicht OK ist, wird die Bestellung abgewiesen.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 38

UML - Aktivitätsdiagramm

(Beispiel zur Spezifikation des Anwendungsfalles „Bestelleingang“)



Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 41

UML - Aktivitätsdiagramm (Bestelleingang und Nachlieferung)

Homogene Spezifikation, aber

- was passiert bei vielen offenen Bestellungen?
- wie passiert die Kommunikation zwischen einem Ablauf des Typs Nachlieferung und vielen Nachbestellungen?

Allgemein: Wie legt ein Ablaufmodell (ausgedrückt als Aktivitätsdiagramm) die Kommunikation auf der Ebene einzelner Abläufe fest?

Zur Erinnerung: OOA dient der Spezifikation; es müssen keine vollständigen Festlegungen der Implementierungen entstehen. Es ist deshalb durchaus nützlich, mit Beschreibungen zu arbeiten, die nicht alles festlegen, solange man sich der Lücken bewusst ist.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 42

UML - Aktivitätsdiagramm (Strukturierung)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

Das zusammengefügte Beispiel (Bestelleingang und Nachlieferung) ist nicht eindeutig einer Klasse, einer Operation oder einem Anwendungsfall zuzuordnen (weil es durch Komposition zweier getrennter Abläufe entstanden ist).

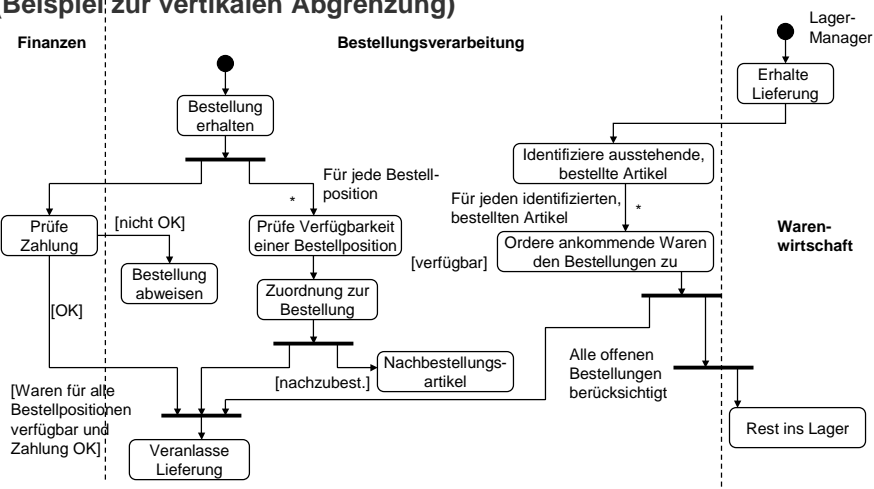
Da man einerseits die Wechselwirkungen beschreiben möchte (deshalb die Komposition) und andererseits die klare Zuordnung nicht aufgeben möchte, gibt es das Strukturierungsmittel der vertikalen Abgrenzung.

Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 43

UML-Aktivitätsdiagramm (Beispiel zur vertikalen Abgrenzung)

UNIVERSITÄT LEIPZIG

LPZ  E-BUSINESS
applied telematics

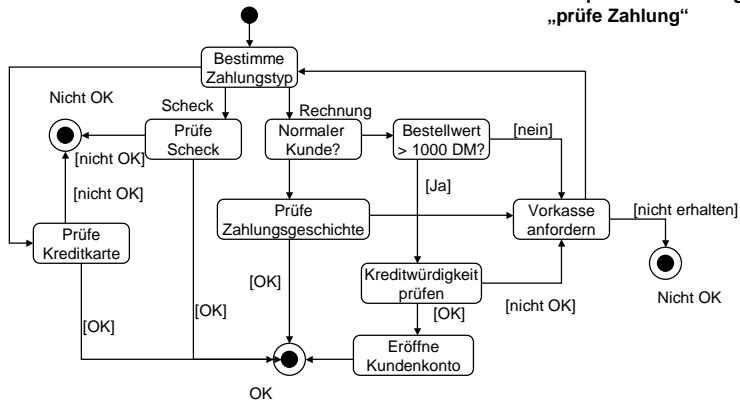


Digitale Informationsverarbeitung WS02/03 ▪ 09.01.2003 ▪ Folie 44

UML-Aktivitätsdiagramm (Strukturierung)

Aktivitäten können verfeinert werden
(passende Abstraktionsebenen für
unterschiedliche Zwecke)

Beispiel: Verfeinerung der Aktivität
„prüfe Zahlung“



UML - Aktivitätsdiagramme (Bewertung)

geeignet für die Modellierung von Geschäftsprozessen über die Grenzen von Anwendungsfällen hinweg.

geeignet für die detaillierte Analyse von Anwendungsfällen.

geeignet für die Modellierung von parallelen Software-Systemen

nicht geeignet für die Beschreibung der Interaktion von Objekten

nicht geeignet für die Zustandsübergänge eines einzelnen Objektes

UML - Komponentendiagramme - Zwischen OOA und OOD

In UML heißt eine Gruppe von „zusammengehörenden“ Klassen *package* (Komponente).

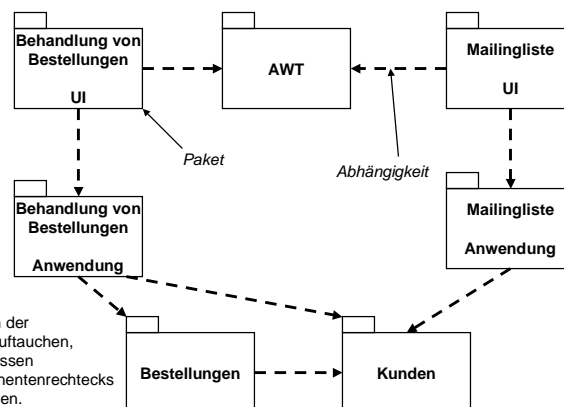
Die Zusammenfassung von Klassen zu Komponenten dient zur Strukturierung von Klassendiagrammen. Oft ist diese Strukturierung ein erster Entwurfsschritt.

Als Hilfe dient die Ermittlung von Abhängigkeiten.

Zwei Komponenten sind abhängig, wenn Änderungen an der Schnittstelle der einen der anderen bekannt gemacht werden müssen.

Zyklische Abhängigkeiten sind in UML nicht verboten, sollten aber dringend aufgelöst werden.

Komponentendiagramme - Zwischen OOA und OOD



Im kleinen Rechteck links oben, kann dann der Komponentename auftauchen, wenn die internen Klassen innerhalb des Komponentenrechtecks angezeigt werden sollen.