



<TITEL DES PRAKTIKUMS>

<TITEL DES TEILPROJEKTES>

Software-Entwicklungspraktikum (SEP)
Sommersemester 2017

Technischer Entwurf

Auftraggeber
Technische Universität Braunschweig
<Name des Instituts>
<Name des Institutsleiters>
<Straße und Hausnummer>
<Postleitzahl und Ort>

Betreuer: <Name>

Auftragnehmer:

Name	E-Mail-Adresse
<Name des Teilnehmers>	<E-Mail-Adresse>
<Name des Teilnehmers>	<E-Mail-Adresse>
<Name des Teilnehmers>	<E-Mail-Adresse>
...	...

Braunschweig, 29. März 2017

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1
1.1
2
2.1
3
3.1
3.2
3.3
4
5
5.1
6
6.1
6.2
7
8
9
9.1
9.2
9.3
10

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

Inhaltsverzeichnis

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

Abbildungsverzeichnis

1 Einleitung

Hier erfolgt eine kurze Darstellung von Aufbau und Ziel dieses Dokuments.

Hier ist die Arbeitsweise des Systems anhand von Aktivitätsdiagrammen und/oder Statecharts darzustellen und kurz verbal beschreiben.

Beispiele:

- Bei einem Spiel könnte der Ablauf des Spiels als Aktivitätsdiagramm dargestellt werden.
- In einen Web-System könnten die vom Nutzer sichtbaren Seiten als Zustände im Statechart modelliert werden.

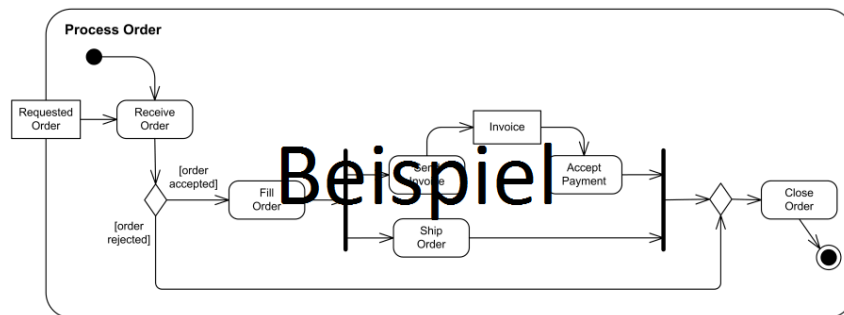


Abbildung 1.1: Ein beispielhaftes Aktivitätsdiagramm

Quelle der Abbildung ??: <http://www.uml-diagrams.org/>

Hinweis zu den Templates:

Dieses Template enthält Hinweise und Beispiele, die selbstverständlich zu entfernen sind. Angaben in <...> sind mit dem entsprechendem Text zu füllen. Die Grafiken in diesem Dokument sind nur Beispiele und sind keine Vektorgrafiken.

Aufgabe des Technischen Entwurfs:

Der Technische Entwurf dokumentiert die klassischen Entwurfsentscheidungen wie z.B. die Verwendung bestimmter Bibliotheken oder Entwurfsmuster. Darüber hinaus bildet der Technische Entwurf die Grundlage der Implementierung, d.h. anhand dieses Dokumentes muss jeder Softwareentwickler in der Lage sein, das Produkt zu entwickeln. Es ist also auf Vollständigkeit der Dokumentation zu achten.

Kapitel die bereits im Fachentwurf bearbeitet wurden, müssen hier nicht erneut

bearbeitet werden. Es sollten jedoch die Annotationen umgesetzt worden sein. Es sollen also Kapitel 3-5 und Kapitel 8-9 neu erarbeitet werden. Kapitel 10 soll an dieses Dokument angepasst werden, wenn es notwendig ist.

Dieses Kapitel kann aus dem Fachentwurf übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten.

1.1 Projektdetails

Besonders interessante oder komplizierte Sachverhalte sollen hier noch weiter vertieft werden. Auch hier sollen wieder Aktivitätsdiagramme und Statecharts verwendet werden.

Beispiele:

- Bei einem Spiel könnten komplizierte Regeln dargestellt werden.
- In einen Web-System könnte bestimmte Workflows als Aktivitätsdiagramm dargestellt werden.

Es kann pro Sachverhalt ein Abschnitt hinzugefügt werden.

2 Analyse der Produktfunktionen

Dieses Kapitel kann aus dem Fachentwurf übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten.

In diesem Kapitel wird das Verhalten für die einzelnen Produktfunktionen analysiert. Dies geschieht, um später eine geeigneten Architektur realisieren zu können, auf Basis der im Pflichtenheft analysierten Produktfunktionen und nicht-funktionalen Anforderungen, die realisiert werden müssen. Jede betrachtete Funktion wird in einem eigenen Unterkapitel dokumentiert. Fügen Sie bitte so viele Unterkapitel ein, wie Produktfunktionen im Pflichtenheft vorhanden sind ein. Auch die nicht-funktionalen Anforderungen sind, so weit möglich, entsprechend darzustellen.

2.1 Analyse von Funktionalität <ID aus Pflichtenheft>: <Funktionsname>

Z.B.: Analyse von Funktionalität **F20**: Automatisches Einlagern

In diesem Abschnitt wird jede Produktfunktion analysiert und als Sequenzdiagrammen dargestellt. Die Analyse umfasst dabei eine Beschreibung, welche Nachrichten die einzelnen Komponenten des Systems austauschen und welche Funktionen aufgerufen werden um die jeweilige Funktion zu realisieren. Anschließend erfolgt die Darstellung der Realisierung der Funktion als Interaktion von Komponenten des zu entwickelnden Systems in einem Sequenzdiagramm. Das Diagramm sollte, so es nicht bereits in der Analyse enthalten ist, detailliert beschrieben werden.

Quelle der Abbildung ??: <http://www.uml-diagrams.org/>

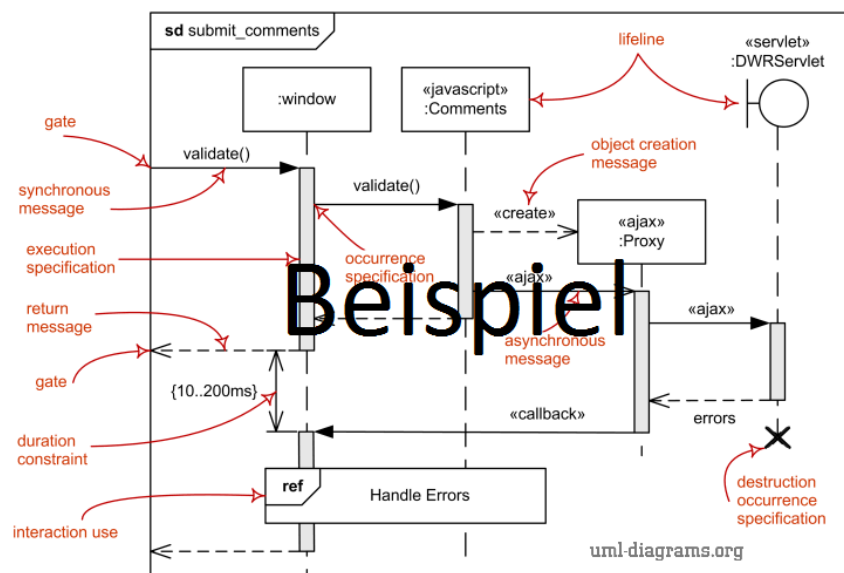


Abbildung 2.1: Ein beispielhaftes Sequenzdiagramm

3 Resultierende Softwarearchitektur

Dieser Abschnitt hat die Aufgabe, einen Überblick über die zu entwickelnden Komponenten und Subsysteme zu liefern.

3.1 Komponentenspezifikation

In diesem Abschnitt wird die aus der Analyse der Produktfunktionen (??) resultierende Komponentenstruktur durch ein Komponentendiagramm beschrieben. Die Bezeichnungen und Anzahl der Komponenten sollten natürlich konsistent zu denen aus ?? sein. Gab es Änderungen, so sollen diese in Kapitel 9 beschrieben werden.

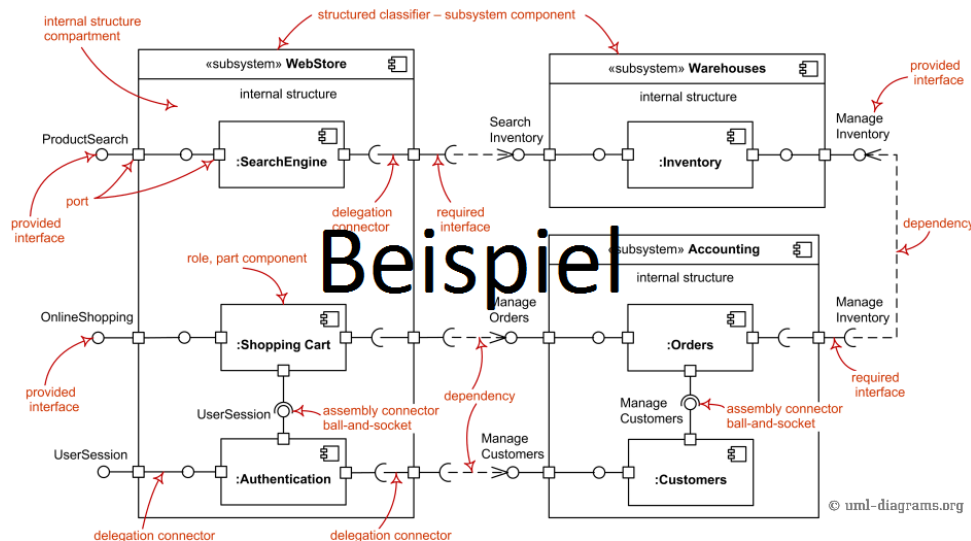


Abbildung 3.1: Ein beispielhaftes Komponentendiagramm.

Quelle der Abbildung ??: <http://www.uml-diagrams.org/>

Die einzelnen Komponenten werden anschließend kurz beschrieben.

Komponente <C10>: <Bezeichnung>

Aufgaben der Komponente

Komponente <C20>: <Bezeichnung>

Aufgaben der Komponente

3.2 Schnittstellenspezifikation

Im Folgenden werden die einzelnen Schnittstellen der Komponenten aus der Komponentenspezifikation näher erläutert, d.h. die von ihnen zur Verfügung gestellten Operationen werden dokumentiert. Die Tabelle ist dabei um so viele Zeilen zu erweitern, wie es Schnittstellen im Komponentendiagramm gibt. In der innen liegenden Aufteilung ist für jede Operation einer Schnittstelle eine Zeile einzufügen. Reine Set- und Get-Aufrufe brauchen nicht aufgeführt zu werden (sollten auch möglichst nicht komponentenübergreifend auftauchen).

Schnittstelle <I10>: <Bezeichnung>

Operation	Beschreibung
<Signatur der Operation>	<Aufgabenbeschreibung der Operation>

3.3 Protokolle für die Benutzung der Komponenten

In diesem Abschnitt wird mit Hilfe von Protokoll-Statecharts die korrekte Verwendung der zu entwickelnden Komponenten dokumentiert. Dies ist insbesondere für diejenigen Komponenten notwendig, für die eine Wiederverwendung möglich erscheint oder sogar bereits geplant ist.

Begründen Sie, für welche Komponenten eine Wiederverwendung sinnvoll erscheint und für welche nicht!

Fügen Sie so viele Statechartdiagramme ein, wie sie Komponenten gefunden haben.

4 Verteilungsentwurf

Sollte es sich bei dem Produkt um eine verteilte Anwendung handeln, so wird diese in diesem Abschnitt dokumentiert. Die Verteilung der Komponenten auf die beliebigen Knoten wird durch das folgende Verteilungsdiagramm beschrieben.

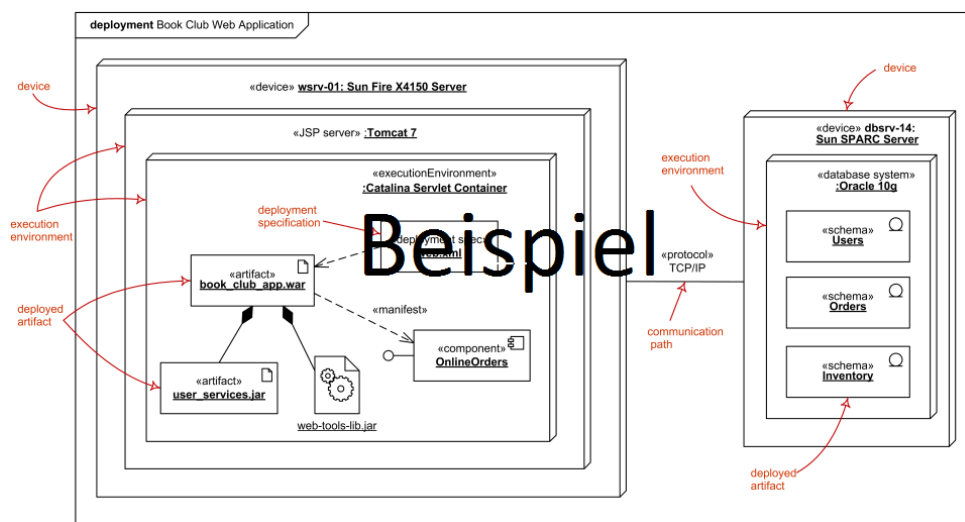


Abbildung 4.1: Ein beispielhaftes Verteilungsdiagramm

Quelle der Abbildung ??: <http://www.uml-diagrams.org/>

5 Implementierungsentwurf

Dieser Abschnitt hat die Aufgabe, alle verwendeten Klassen und Bibliotheken zu dokumentieren. Dabei wird jede Komponente aus ?? gesondert betrachtet. Für Entwurfsentscheidungen, die mehr als eine Komponente betreffen, wird mit Verweisen zwischen den Dokumentationen der Komponente gearbeitet. Es sind dabei so viele Unterabschnitte einzufügen, wie Komponenten vorhanden sind. Die Anzahl sollte also mit den Komponenten aus ?? übereinstimmen.

5.1 Implementierung von Komponente <ID>: <Komponentenname>:

z.B.: Implementierung von Komponente ??: <Komponentenname>

Beschreiben Sie hier bitte die Implementierung der Komponente. Erläutern Sie dabei bitte, welche Entwurfsmuster und Bibliotheken Sie verwenden. Die Implementierung wird dabei durch Klassendiagramme dokumentiert.

5.1.1 Paket-/Klassendiagramm

Hier werden die Klassen der Komponente dargestellt. Denken Sie an die Pakete

Quelle der Abbildung ??: <http://www.uml-diagrams.org/>

5.1.2 Erläuterung

Die verwendeten Attribute, Aufgaben und Kommunikationspartner sind für jede Klasse kurz zu erläutern. Die ankommenden Nachrichten beziehen sich dabei auf die Sequenzdiagramme der Feinanalyse im Fachlichen Entwurf und stellen meist aufzurufende Methoden der Klasse dar. Reine Get-/Set-Methoden oder Bibliotheksfunktionen brauchen nicht aufgeführt zu werden.

Beispiel:

ConnectionManager<CL10>

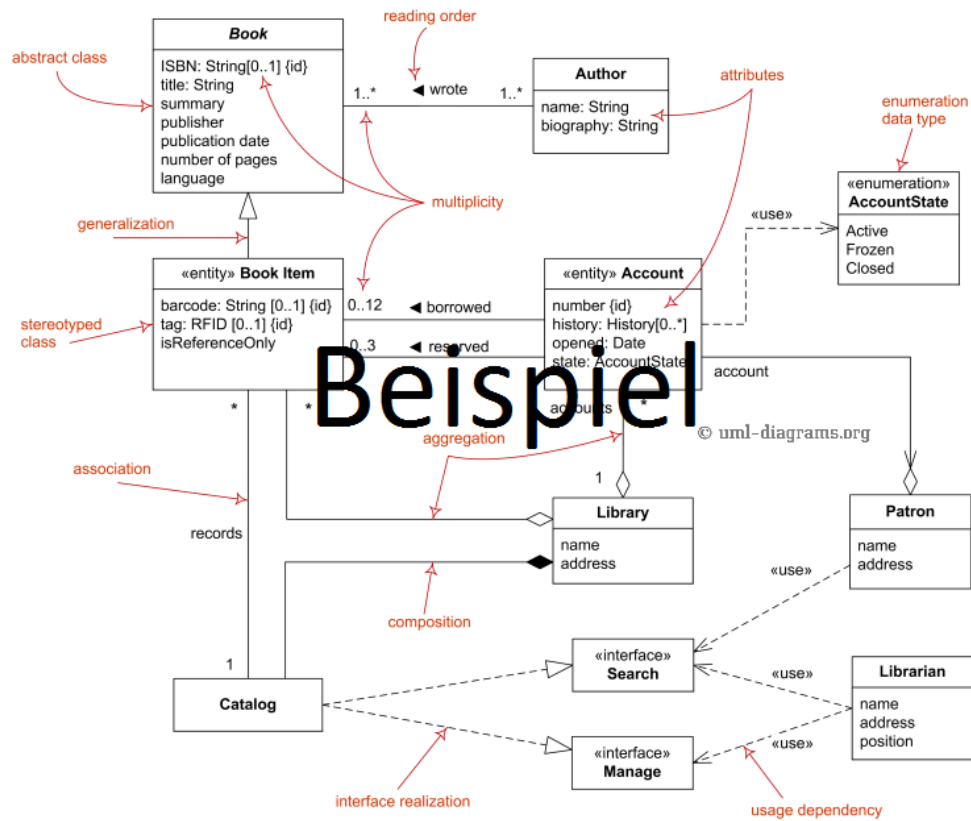


Abbildung 5.1: Ein beispielhaftes Klassendiagramm für Komponente ??

Aufgabe

Verwaltung der Verbindungen zum Server

Attribute

kein

Operationen

connect(): Diese Operation baut ein Verbindung zum Server auf

Kommunikationspartner

Server

Alternative: Gibt es in eurem Projekt eine Dokumentation des Programmcodes mit Java-Doc/Doxygen oder Ähnlichem, dürft ihr diese in den Anhang tun und hier darauf verweisen.

6 Datenmodell

Dieses Kapitel kann aus dem Fachentwurf übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten.

Falls in der Anwendung bestimmte Daten dauerhaft gespeichert werden, so sind die entsprechenden Entities und Beziehungen hier darzustellen und zu erläutern. Dieses Kapitel ist relevant, sobald Daten dauerhaft in einer (relationalen) Datenbank, einer Datei oder ähnlichem abgespeichert werden. Diese Daten sollten mit den im Pflichtenheft bereits beschriebenen Daten übereinstimmen oder beschrieben werden, wieso diese Daten nicht mehr erforderlich sind.

XML-Dateien: Benutzt ihr das Dateiformat XML, so solltet ihr hier auch das XML Schema beschreiben, sofern ihr eines habt. Gleiches gilt natürlich für andere Dateiformate, die ähnliche Konzepte nutzen.

6.1 Diagramm

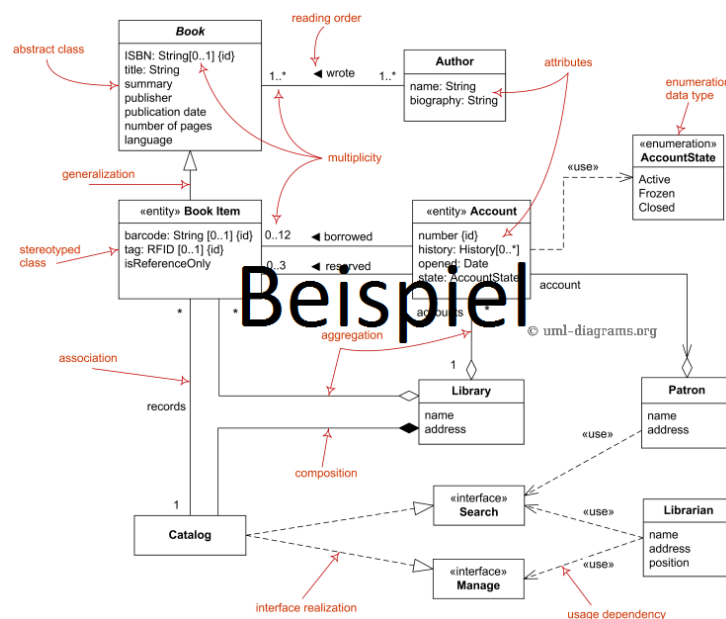


Abbildung 6.1: Ein beispielhaftes Klassendiagramm

Quelle der Abbildung ??: <http://www.uml-diagrams.org/>

6.2 Erläuterung

Die Tabelle ist um so viele Einträge zu erweitern, wie es Entities im obigen Klassendiagramm gibt. Für jede Entity sind so viele Einträge in der Beziehungs-Tabelle einzufügen, wie es Beziehungen zu dieser Entity gibt.

Wurden aus Platzgründen Attribute im Klassendiagramm nicht aufgeführt, so sollte diese hier beschrieben werden.

Benutzer ⟨E10⟩

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
<Name der Beziehung>	<Kardinalität>	Min: <...>, Max:<...>	<Beschreibung>

Artikel ⟨E20⟩

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
<Name der Beziehung>	<Kardinalität>	Min: <...>, Max:<...>	<Beschreibung>

7 Konfiguration

Dieses Kapitel kann aus dem Fachentwurf übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten.

Sollte für die Bearbeitung und Nutzung des Produktes ein Rechner/Server mit einer bestimmten Konfiguration erforderlich sein, so ist hier dessen Konfiguration zu beschreiben. Dies geschieht durch explizite Nennung aller Konfigurationsdateien und notwendiger Einträge.

Gibt es in eurem Projekt also config-Dateien so sollten diese hier alle aufgezählt werden. Es sollte beschrieben werden, wo diese zu finden sind und welchen Zweck sie erfüllen. Gibt es Parameter die angepasst werden können, so sollte auf diese eingegangen werden.

8 Änderungen gegenüber Fachentwurf

In diesem Kapitel sollt ihr Änderungen beschreiben, die gegenüber dem Fachentwurf geschehen sind. Es kann euch z.B. aufgefallen sein, dass ihr eine weitere Komponente benötigt, die ihr in Kapitel ?? nicht in den Sequenzdiagrammen beschrieben habt. Ihr solltet dazu dann die Sequenzdiagramme anpassen und hier erneut beschreiben. Gleiches gilt für die Kapitel Datenmodell und Konfiguration.

9 Erfüllung der Kriterien

Nachfolgend wird beschrieben, wie die einzelnen Kriterien des Pflichtenheftes erfüllt werden und worauf geachtet wird. Es ist dabei explizit auf die definierten Kriterien des Pflichtenheftes zu verweisen. Das Kapitel sollte fortlaufend gepflegt werden. **Es sollte auch darauf verwiesen werden, welche Komponente das Kriterium erfüllt.**

9.1 Musskriterien

Die folgenden Kriterien sind unabdingbar und müssen durch das Produkt erfüllt werden:

RM1 wird durch folgende Komponente umgesetzt.

RM2 ...

9.2 Sollkriterien

Die Erfüllung folgender Kriterien für das abzugebende Produkt wird angestrebt:

RS1 ...

RS2 ...

9.3 Kannkriterien

Die Erfüllung folgender Kriterien für das abzugebende Produkt wird angestrebt:

RC1 ...

RC2 ...

10 Glossar

Hier werden Fachbegriffe erklärt.