Hinweise zum Bearbeiten des Entwurfsdokuments

(Im Rahmen der Veranstaltung Modellierung II, SoSe 2016)

Allgemeine Hinweise:
Alles, was in dieser Schriftart gesetzt ist, dient nur zur Erläuterung und sollte im fertigen Dokument nicht mehr auftauchen.

Im Dokument können (und sollen) Diagramme und verschiedene Abbildungen verwendet werden. Diagramme und Abbildungen ohne erläuternden Text sind jedoch wertlos. Bitte achten Sie daher darauf, dass jede Abbildung im Text referenziert und auch erklärt wird. Das Verhältnis zwischen erläuternden Texten und Abbildungen sollte möglichst ausgewogen sein, d.h. ein Dokument mit 40 Seiten darf maximal 20 Seiten Abbildungen enthalten. Dieses Dokument soll insgesamt maximal 40 Seiten umfassen.

Das Template soll vollständig bearbeitet werden

Latex-Hinweis:
Bilder werden folgendermaßen referenziert: In Abbildung **??** *steht ...*

Entwurfsdokument

(Template zur Veranstaltung Modellierung II, SoSe 2016)

Projekt: *Name des Projekts*

Auftraggeber:*Anschrift Auftraggeber*

Auftragnehmer:*Anschrift Auftragnehmer*

Verantwortlichkeit	Name, Vorname	Datum
Ansprechpartner		
Bearbeitender		
Bearbeitender		
Bearbeitender		

1 Abstrakte Architektur

Dieser Abschnitt beschreibt im Wesentlichen eine grobe Zerlegung der zu entwickelnden Anwendung in (Teil-)Komponenten und deren Zusammenhänge. Die Struktur der einzelnen Komponenten (also die enthaltenen Klassen und deren Beziehungen) sowie deren Schnittstellen ist nicht Teil dieser Beschreibung. Die notwendigen Schnittstellen werden erst im Laufe dieses Dokumentes entwickelt.

Die hier definierten Komponenten sind als logische Einheiten zu verstehen, die aus beliebig vielen Klassen bestehen können, die gemeinsam die Funktionalität der Komponente realisieren.

Das folgende Komponentendiagramm führt (Teil-)Komponenten ein und beschreibt ihre Abhängigkeiten.

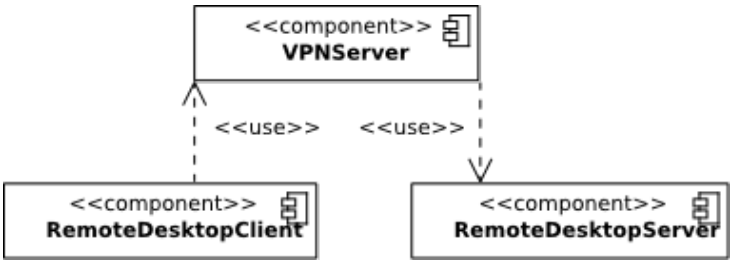

Abbildung 1: Durch eigenes Komponentendiagramm ersetzen

2 Interaktion der Komponenten

Die Interaktion zwischen den Komponenten während der Ausführung eines Use Case wird durch Sequenzdiagramme (oder Interaktionsdiagramme) auf Instanzen der Komponenten (repräsentiert durch Instanzen von Schnittstellenklassen/Interfaces) beschrieben. Die dabei verwendeten Operationen müssen sich später in den Schnittstellen der Komponenten wiederfinden.

Es müssen nicht alle Use Cases hier beschrieben werden. Stattdessen sollen nur wesentliche oder charakteristische Use Cases (z. B. solche, deren Ablauf bei anderen Use Cases sehr ähn-lich stattfindet) beschrieben werden.

Interaktion bei Ausführung von
<Use Cae-ID>: <Use Case-Name>

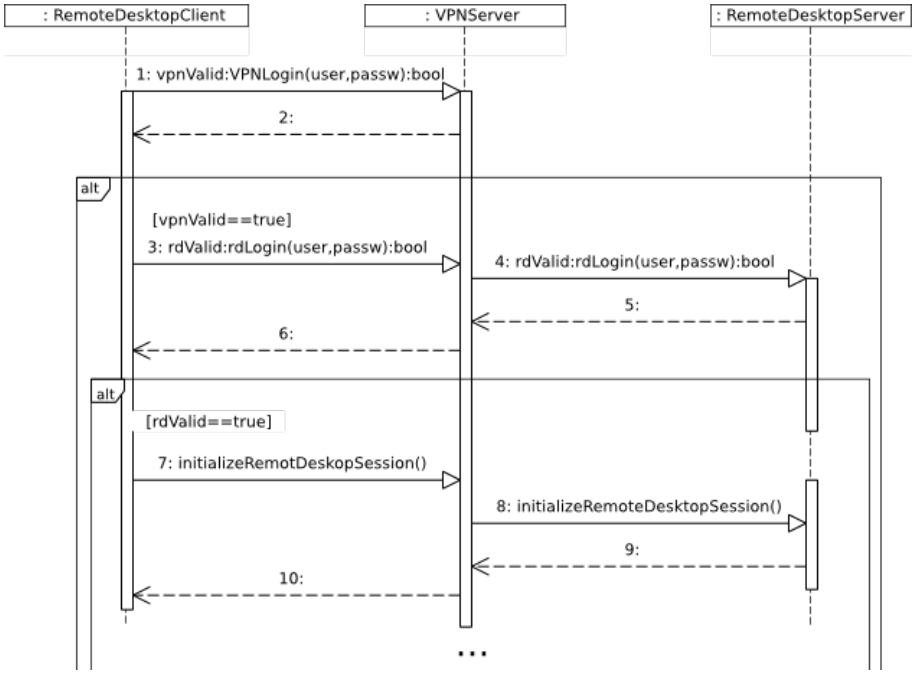

Abbildung 2: Durch eigenes Sequenzdiagramm ersetzen

3 Komponentenschnittstellen

Hier sollen die Schnittstellen der in Abschnitt 1 eingeführten Komponenten definiert werden. Des Weiteren sollen hier Nachrichten, die eventuell über die Schnittstellen verschickt werden, definiert werden. Nachrichten sind Klassen, die beispielsweise als Parameter der in den Schnittstellen deklarierten Operationen verwendet werden.

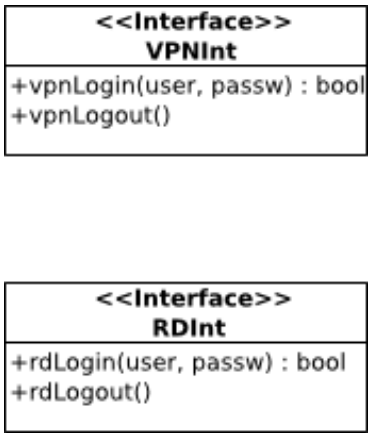

Abbildung 3: Durch eigenes Klassendiagramm ersetzen

4 Konkrete Architektur

In diesem Abschnitt werden die zuvor identifizierten Schnittstellen der Komponenten genauer beschrieben. Im Komponentendiagramm werden daher Abhängigkeiten und Implementierungsbeziehungen beschrieben bzw. ergänzt.

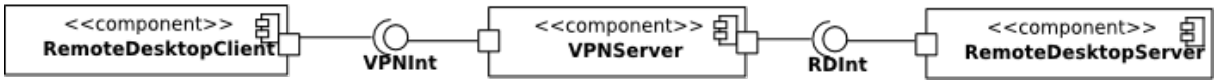

Abbildung 4: Durch eigenes Komponentendiagramm ersetzen

5 Komponenten

Es sollen die in der Architektur definierten Komponenten weiter beschrieben werden. Dazu ist ihre innere Struktur zu beschreiben und mit Hilfe von Klassendiagrammen, Kompositionsstrukturdiagrammen sowie Komponentendiagrammen zu modellieren. Pro Komponente sollen Parts, Ports und Konnektoren beschrieben werden. Pro Komponente sollen Klassen und Assoziationen sowie deren Methoden und Attribute definiert werden, die für die Durchführung der vorher beschriebenen Abläufe innerhalb der Komponenten benötigt werden. In den Klassendiagrammen müssen sich gegebenenfalls die in Abschnitt 4 definierten Schnittstellen wiederfinden.

Komponenten können sich während der Ausführung in verschiedenen Zuständen befinden. Sie wechseln diese, ausgelöst durch interne oder externe Ereignisse. Zustandsbezogenes Verhalten wird durch Zustandsdiagramme modelliert. Auslösende Ereignisse müssen als Operationen an den Komponentenschnittstellen zur Verfügung stehen. Ebenso sollen durch Transitionen ausgelöste Aktionen in der Komponente oder an externen Komponenten verfügbar sein.

Es muss nicht für jede Komponente ein Zustandsdiagramm erstellt werden. Zustandsdiagramme sollen nur für die Komponenten modelliert werden, die reaktiven Charakter haben und in verschiedenen Modi operieren.

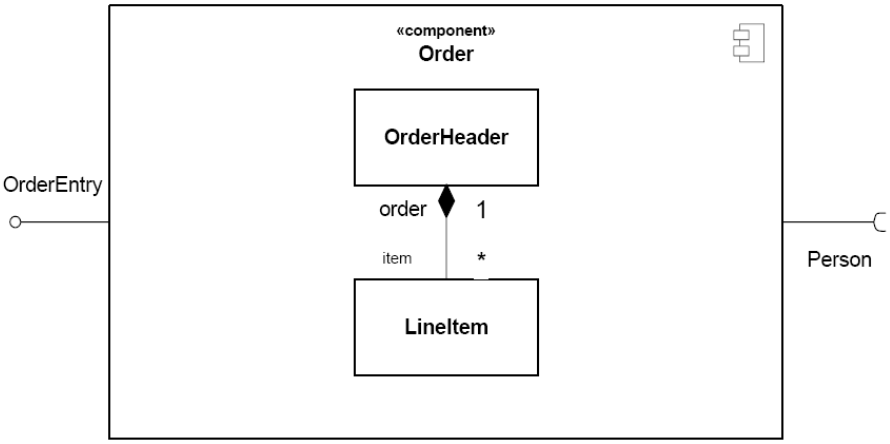

Abbildung 5: Durch eigene Diagramm ersetzen

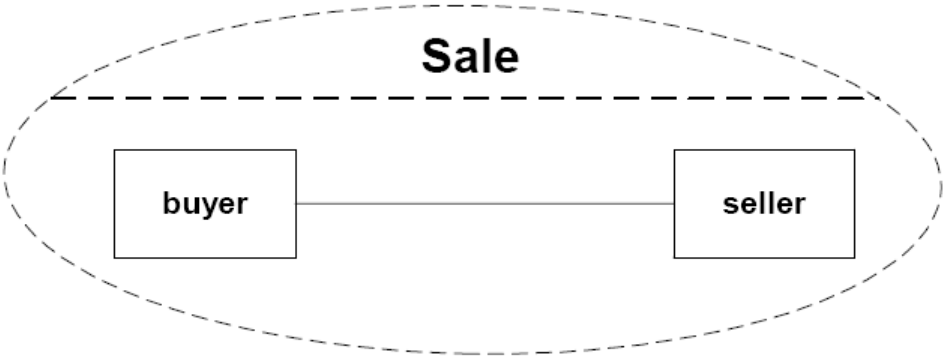

Abbildung 6: Durch eigene Diagramm ersetzen

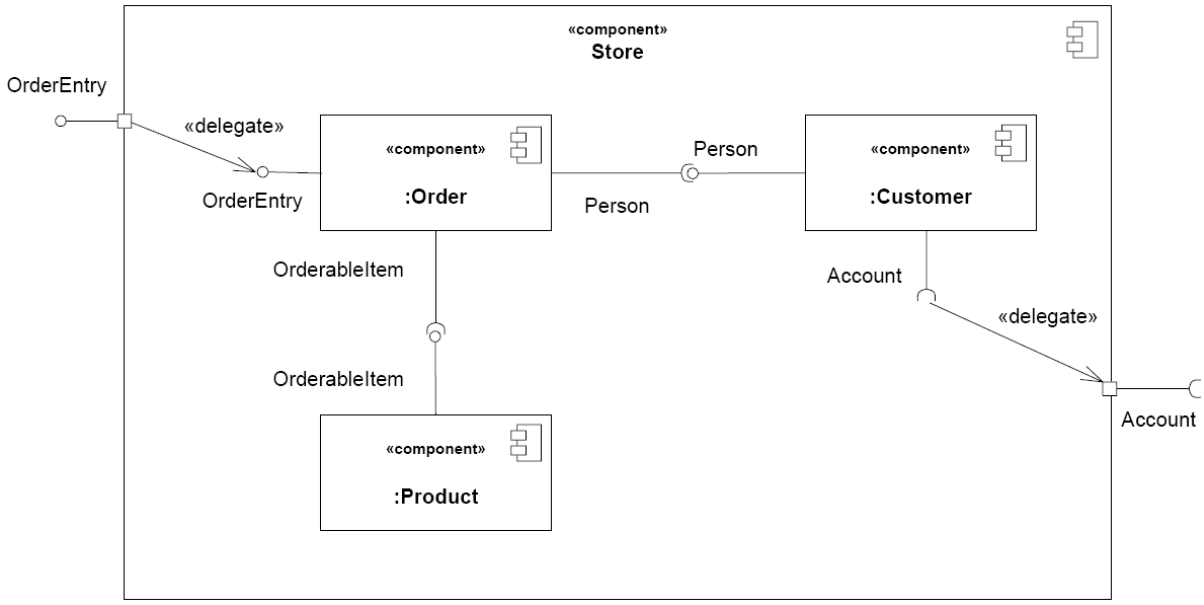

Abbildung 7: Durch eigene Diagramm ersetzen

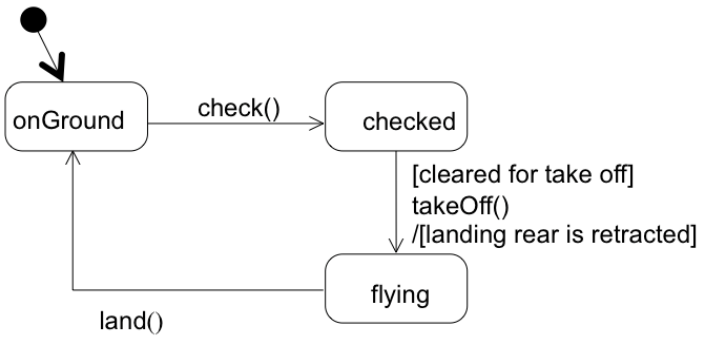

Abbildung 8: Durch eigene Diagramm ersetzen

6 Paketstruktur

Dieser Abschnitt beschreibt im Wesentlichen eine grobe Zerlegung der zu entwickelnden Anwendung in Pakete und deren Zusammenhänge. Dabei sollen Zusammenhänge und Abhängigkeiten zwischen den Paketen beschrieben werden.

Die hier definierten Pakete sind logische Einheiten oder gemeinsam genutzte Einheiten, die aus beliebig vielen Klassen bestehen können, welche gemeinsam die Funktionalität durch das Paket zusammenfassen. Die Aufteilung von Klassen zu Paketen muss sich dabei nicht an den strukturellen Eigenschaften der Architektur (z.B. Komponenten) orientieren. Vielmehr soll die Zerlegung in Pakete der besseren Strukturierung, Organisation, Arbeitsteilung bei Implementierung und Übersichtlichkeit der in der gesamten Architektur verwendeten Klassen dienen.

Das folgende Paketdiagramm führt Pakete ein und beschreibt ihre Abhängigkeiten.

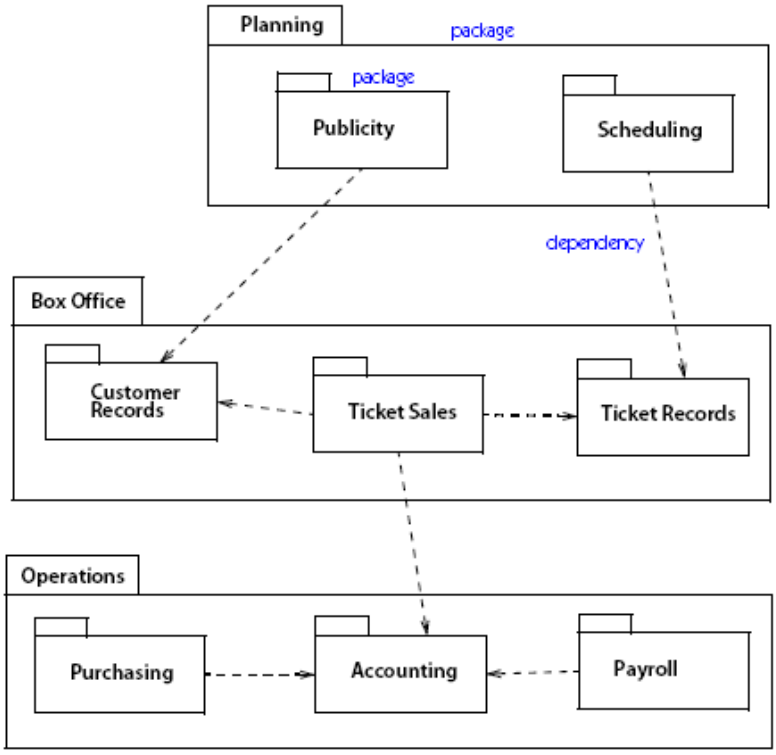

Abbildung 9: Durch eigene Diagramme ersetzen

7 Paketdetails

In diesem Abschnitt sollen die in der Paketstruktur und den vorherigen Abschnitten enthaltenen Klassen und Interfaces weiter detailliert beschrieben werden. Dazu ist die innere Struktur der Pakete mit Hilfe von Klassendiagrammen zu modellieren. Pro Paket sollen Schnittstellen, Klassen und Assoziationen sowie deren Methoden und Attribute detailliert definiert werden. Hierzu gehört z.B. auch die Sichtbarkeit von Attributen und Operationen. Weiterhin sollen die Klassendiagramme an entsprechenden Stellen um OCL-Ausdrücke erweitert werden.

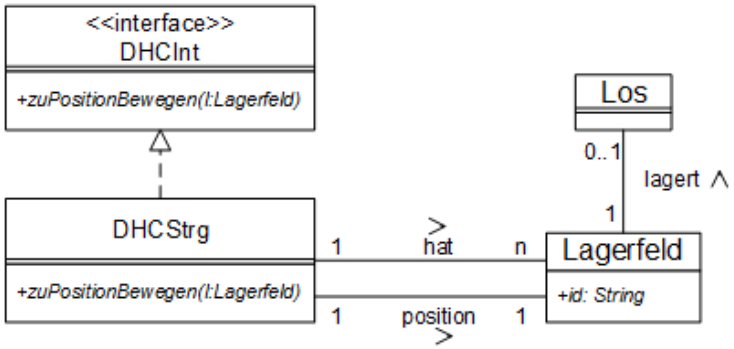

Abbildung 10: Durch eigene Diagramme ersetzen

7.1 Beschreibung der Klasse <Klassenname>

In diesem Abschnitt sollen die in den Klassendiagrammen enthaltenen Klassen/Schnittstellen weiter beschrieben werden. Hierzu soll zu jeder Klasse eine textuelle Beschreibung erstellt werden.

Weiterhin soll für wesentliche Operationen der Ablauf in Form von Aktivitätsdiagrammen modelliert werden. Der Lebenszyklus kann durch ein entsprechendes Zustandsdiagramm modelliert werden

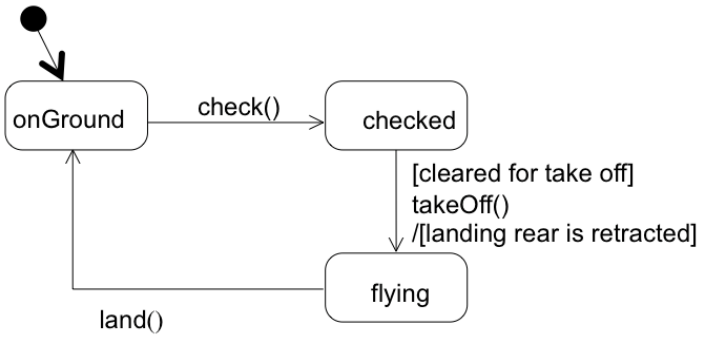

Abbildung 11: Durch eigene Diagramme ersetzen

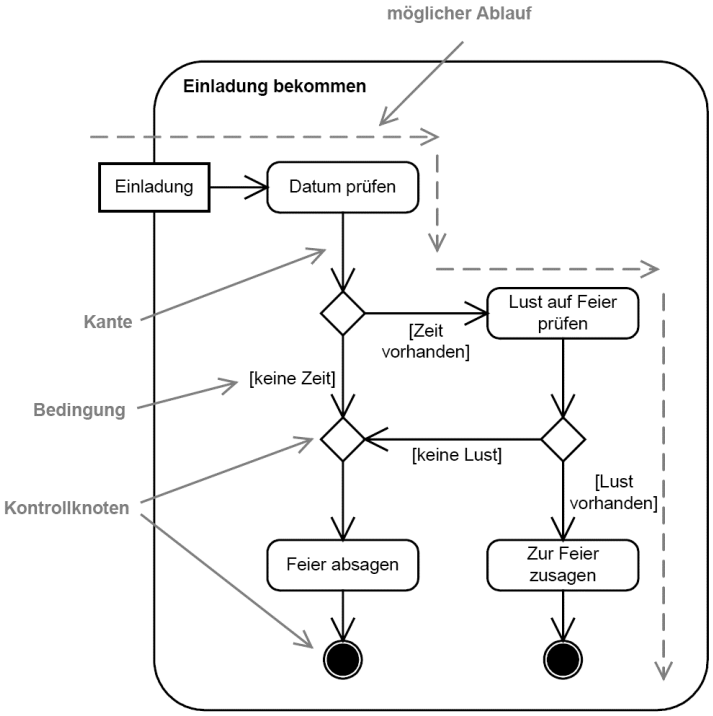

Abbildung 12: Durch eigene Diagramme ersetzen

8 Abläufe

Wesentliche Abläufe innerhalb des Produkts oder innerhalb von Paketen sollen weiter verfeinert werden. Dabei sind gegebenenfalls die Abläufe auf Komponentenebene zu verfeinern sowie das interne Verhalten umzusetzen.

Dabei kann es sich um Abläufe handeln, welche Use Cases entsprechen oder von Objekten einer Klasse selber angestoßen werden (z. B. bei aktiven Klassen).

Die Modellierung kann mit Hilfe von Interaktionsdiagrammen unter Berücksichtigung der in vorangehenden Abschnitten definierten inneren Struktur erfolgen.

Bei der Erstellung der Interaktionen sollen folgende Eigenschaften gelten. Die Rollen der modellierten Interaktionen müssen in den Strukturen vorhanden sein, die verwendeten Operationen der Nachrichten müssen in der zugehörigen Klasse, bzw. Schnittstelle spezifiziert sein, die Interaktion muss mit einem evtl. angegebenen Lebenszyklus (siehe Abschnitt 7.1) zusammenspielen und die umgesetzten Interaktionen sollen Use Cases, bzw. Abläufen auf Komponentenebene entsprechen.

Interaktion <kurzer Titel>
oder
Interaktion bei Ausführung von <Use Case-ID>: <Use Case-Name>

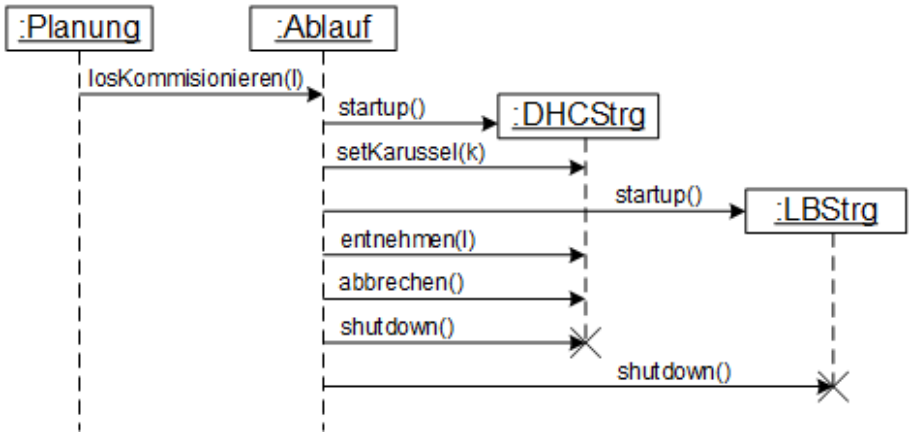

Abbildung 13: Durch eigene Diagramm ersetzen

9 Produkteinsatz

In diesem Abschnitt wird der geplante Einsatz des zu entwickelnden Produktes beschrieben. Dies umfasst insbesondere die Systemumgebung in der das Produkt eingesetzt werden soll und die Zuordnung der Software zu dieser.

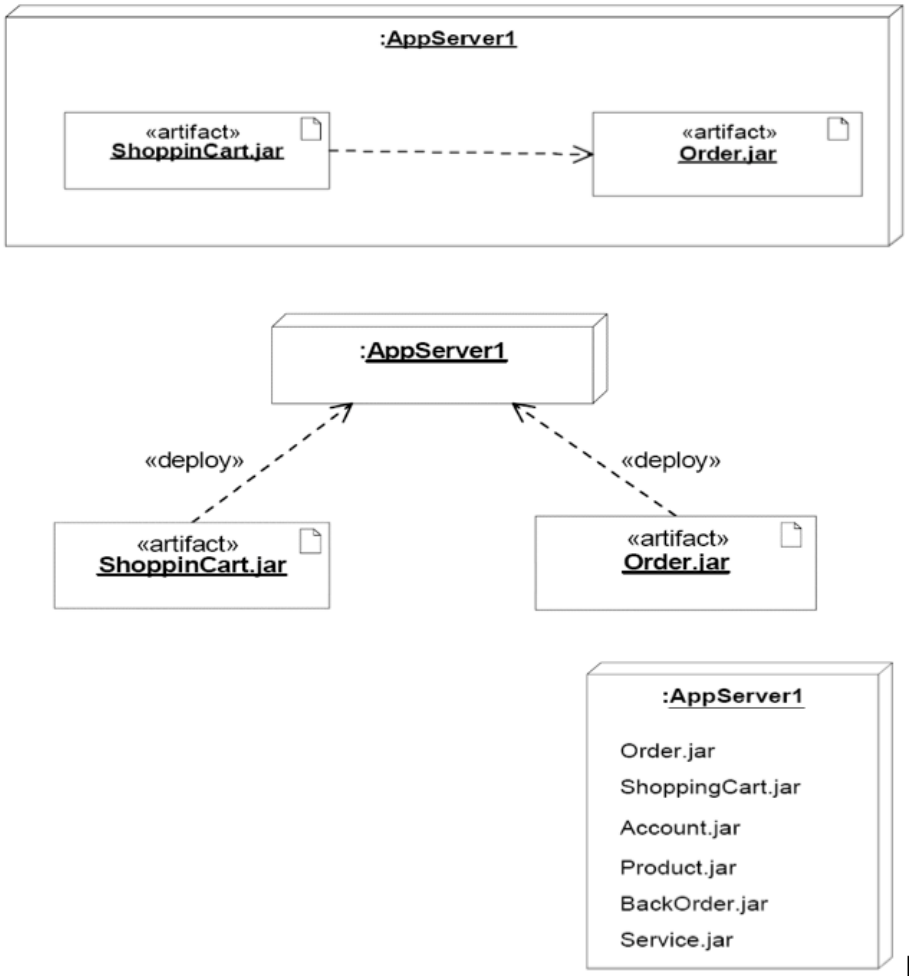

Abbildung 14: Durch eigene Diagramme ersetzen