

11 Komponentendiagramm

Inhalt

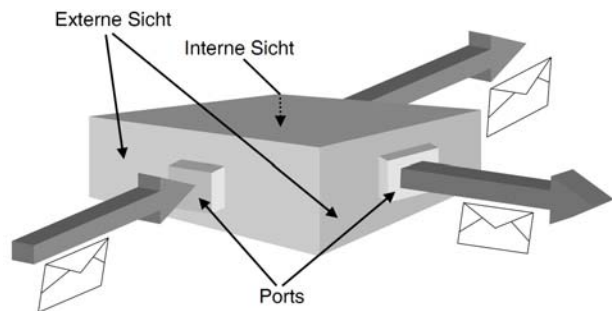
11.1 Allgemein	-2
11.2 Externe und interne Sicht	-3
11.3 Notationsformen	-5
11.4 Verknüpfung von Komponenten	-7
11.5 Beispiele	-9

11.1 Allgemein

- ❑ **Komponentendiagramm** zeigt die Definition von **Komponenten** und **deren Abhängigkeiten**. Damit können einzelne Komponenten, z.B. einer Komponentenbibliothek, oder aber die Auswahl von Komponenten und die entsprechenden Zusammenhänge für eine bestimmte Systemkonfiguration beschrieben werden
- ❑ Definition Komponente (*Component*)
 - eine **modulare Einheit**, die die beinhalteten Elemente kapselt (*self-contained*),
 - mit wohldefinierten **Schnittstellen** (*Interfaces*),
 - wobei diese Einheit innerhalb ihrer Systemumgebung ausgetauscht werden kann (*substitutable*), solange die erforderlichen angebotenen und benötigten Schnittstellen gewährleistet sind (→ **Wiederverwendbarkeit**)
 - entspricht dem Komponentenbegriff in der Softwareentwicklung
 - eine Sonderform einer UML Klasse, wodurch sie alle Klasseneigenschaften wie Attribute, Assoziationen, Vererbungsbeziehungen, Operationen, Interfaces, Ports aufweisen kann und auch instanziiert werden kann
- ❑ Vollständige **Spezifikation der Konzepte siehe [OMG 2015]**; → Literaturangaben!)
- ❑ Hinweis: unterschiedliche Unterstützung durch Tools.

11.2 Externe und interne Sicht

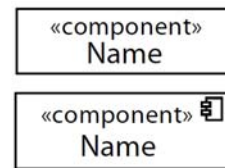
- Externe Sicht (*“black-box” view*), wird bestimmt durch
 - die öffentlich (public) sichtbaren Eigenschaften und Operationen, optional auch ein Verhalten, spezifiziert durch Protokollautomaten (= dynamische Bedingungen zwischen Aufruf von Operationen aus dem Interface)
 - die Verknüpfung einer Komponente mit anderen Komponenten über *Ports*
 - zeigt keine internen Strukturen bzw. Realisierungen der Komponente / Interfaces
- Interne Sicht (*“white-box” view*), wird bestimmt durch
 - eine Spezifikation der Realisierung der angebotenen Interfaces (Details)
 - kann nahezu beliebige Elemente eines Modells enthalten, von Anwendungsfällen bis hin zu physischen Artefakten (Verfeinerung während der Entwicklungszyklen)



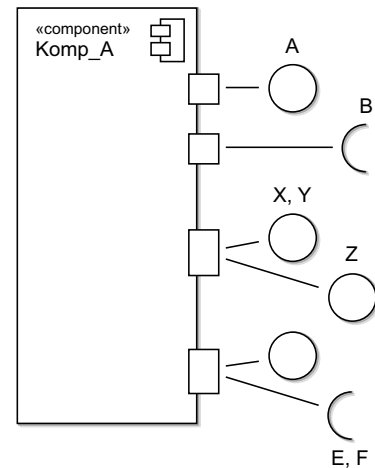
- Zwei häufige Verwendungsformen
 - Spezifikationskomponente - kapselt die externe Sicht
 - Implementierungskomponente - kapselt die interne Sicht

11.3 Notationsformen

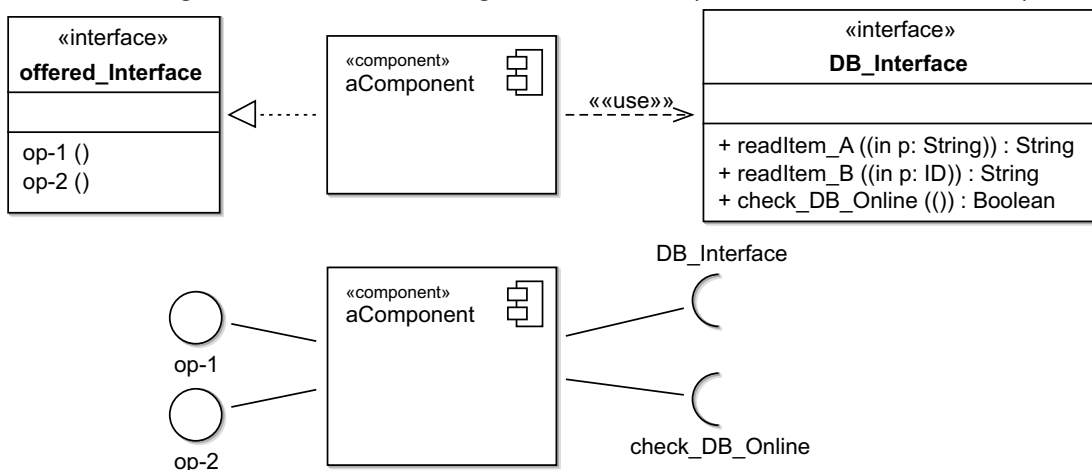
- Notation für Komponente
(mit und ohne Symbol)



- Notation von angebotenen und benötigten Interfaces und deren Zuordnung zu Ports
 - Kreis (*lollipop*): angebotenes Interface
 - Halbkreis (*socket*): benötigtes Interface
 - Rechteck auf Randlinie: Port, kann auch mehrere Interfaces bündeln (Kombinierter Port, *complex port*)

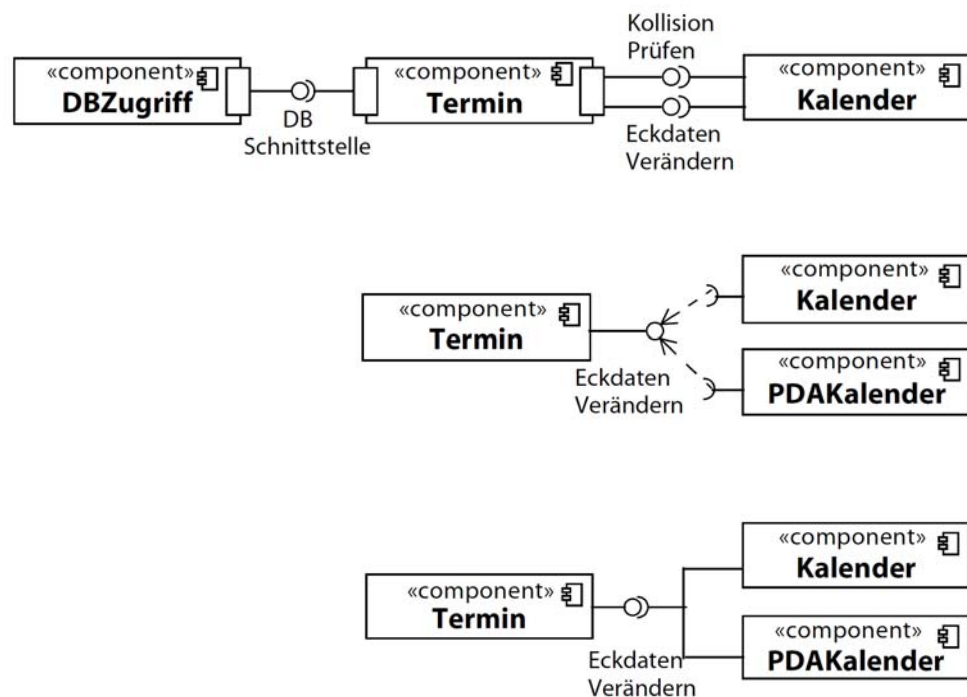
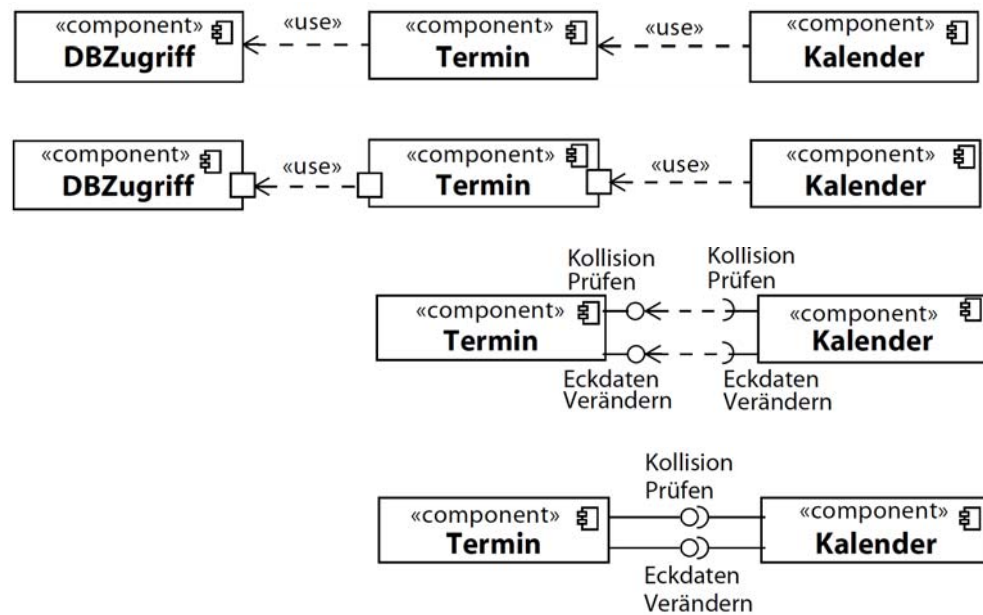


- Notation von angebotenen und benötigten Interfaces (verschiedene Varianten)



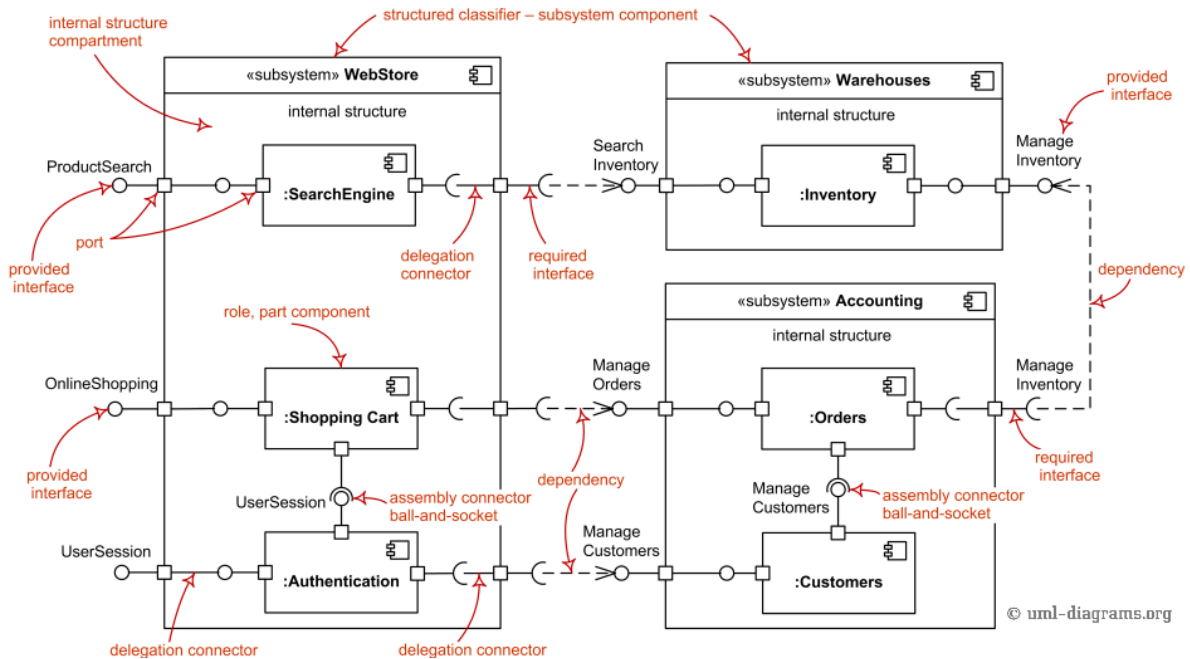
11.4 Verknüpfung von Komponenten

- Ausführliche Erläuterungen zu diesen Beispielen siehe [UML@Work] !



11.5 Beispiele

- Siehe: <http://www.uml-diagrams.org/component-diagrams.html>



Literatur und Quellen:

- [OMG 2015] OMG Unified Modeling Language™ (OMG UML), Version 2.5
Normative Reference: <http://www.omg.org/spec/UML/2.5>
OMG Document Number formal/2015-03-01
- [UML@Work]
Kapitel 3.4 in:
Martin Hitz, Gerti Kappel, Elisabeth Kapsammer, Werner Retschitzegger:
UML @ Work - Objektorientierte Modellierung mit UML2.
dpunkt Verlag 2005 / 3. aktualis. u. überarb. Aufl. 2005.
ISBN-13: 9783898642613 ISBN-10: 3898642615
- Tool zur Vertiefung: BEE-UP Modelling Tool:
<http://austria.omilab.org/psm/content/bee-up/info>
<http://www.omilab.org/web/guest/omilab-in-education/cmmc>