UML 2.0 als Architekturbeschreibungssprache ?

Seminar: Architekturbeschreibungssprachen

Manuel Wickert

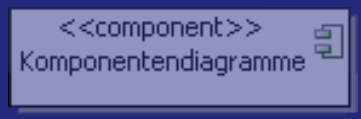
Motivation

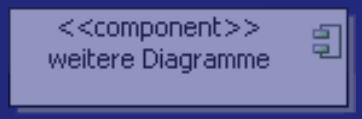
- UML 2.0 nicht als ADL im Sinne von Taylor/Medvidovic entworfen.

- Warum UML als ADL ?
  - weit verbreitet
  - hoher Tool-Support
  - etablierter Standard in der Modellierung von Software
  - seit UML 2.0 näher an einer ADL

Überblick

- Einleitung UML 2.0
- Komponentendiagramme
- weitere Diagramme
  - Sichten
  - Kompositionsstrukturdiagramme
  - Protokoll Zustandsautomat
  - Sequenzdiagramme
- Ausblick / Fazit





UML 2.0

- UML (Unified Modelling Language)
- seit 2004 UML 2.0 Standard
- OMG (Object Management Group)
  - IBM, Apple, Sun
  - CORBA, IDL, MDA
- visuelle Modellierungssprache
- MDA als wichtiges Ziel der UML

- 13 Diagrammtypen

UML 2.0 vs. ADL

- allgemeine Modellierungssprache
- kann viele Aspekte eines Systems modellieren
- Semantik relativ weich um viel Interpretationsraum zu lassen
- nicht nur für Softwaresysteme auch Business Prozesse

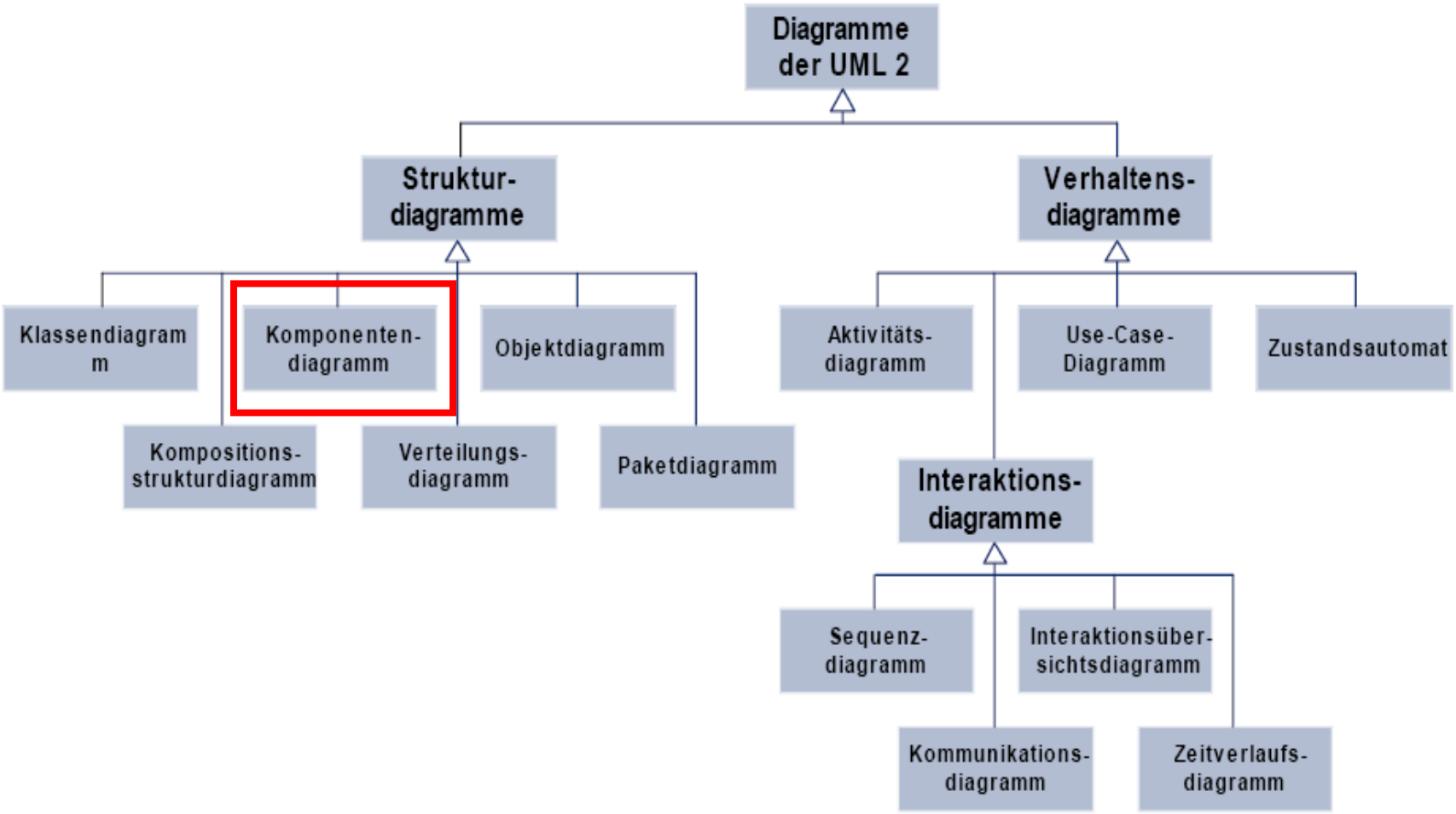
- Ziel → Modellierung von (Objektorientierten) Systemen / MDA

- meist domainspezifisch

- auf die Architektur eines Systems spezialisiert
- Semantik recht präzise um die Simulation zu unterstützen.
- ausschließlich um Hardware-/Softwaresysteme zu modellieren

- Ziel → Architektur modellieren, Simulation, Analyse

Diagramme der UML 2.0

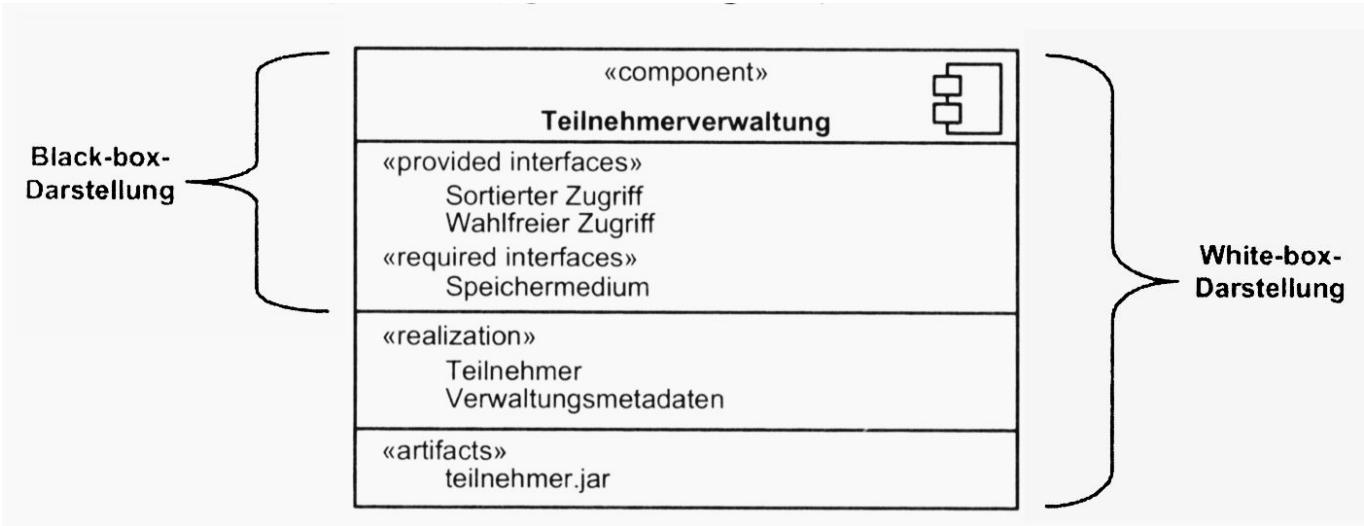


Komponentendiagramme

- komponentenbasierte Entwicklung
- Wiederverwendbarkeit
- Sichtweise näher an der Laufzeit eines Systems
- besteht hauptsächlich aus:
  - Komponenten
  - Schnittstellen
  - Ports
  - Artefakten
  - Klassen
  - Beziehungen
  - (Konnektoren)

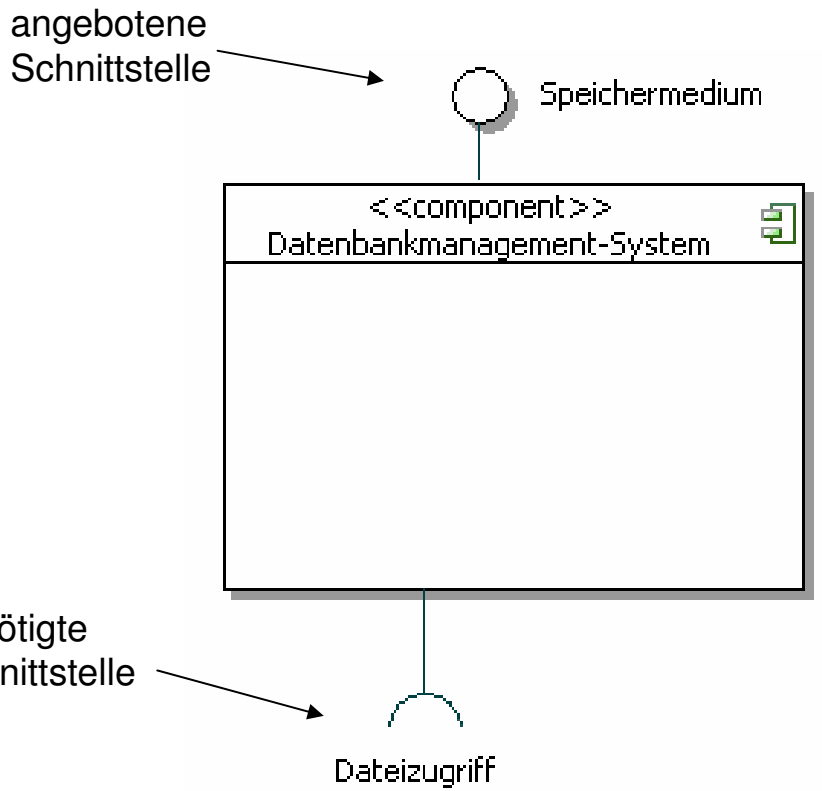
Komponente

- modularer Systemteil
- definiert durch ihre Schnittstellen
- austauschbar in ihrer Umgebung
- Blackbox vs. Whitebox



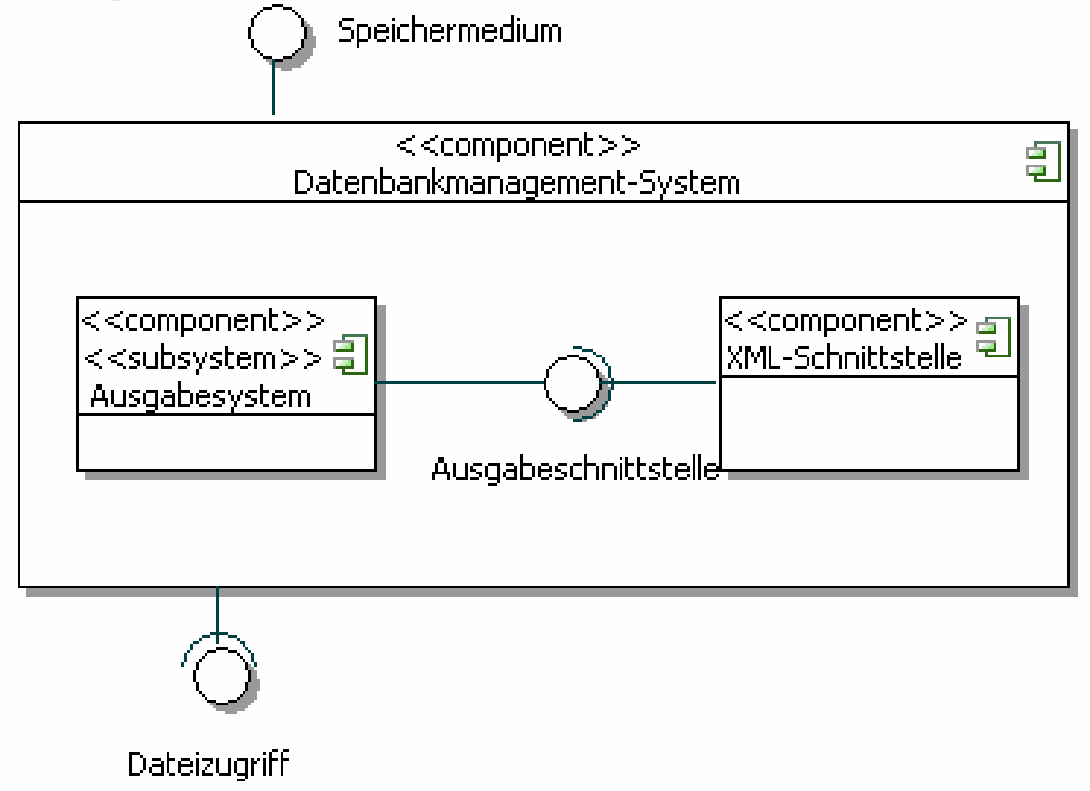
Schnittstelle einer Komponente

- Schnittstelle aus Klassendiagramm
- definiert einen Kontrakt den die implementierenden Komponenten einhalten müssen



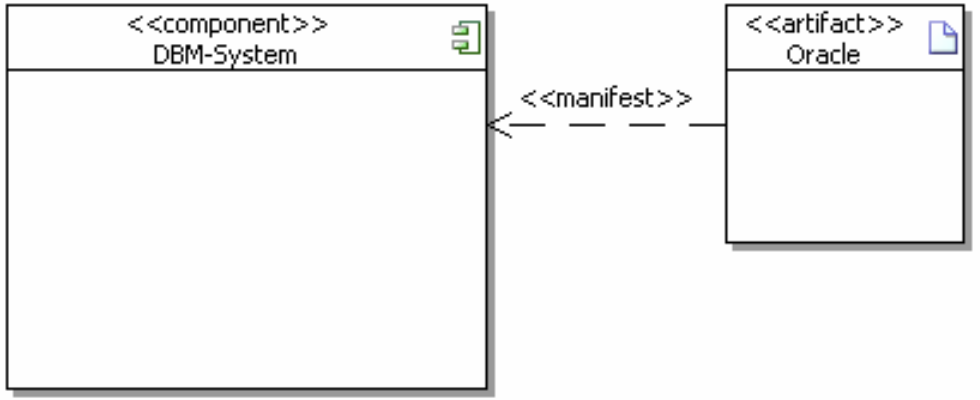
Schachtelung von Komponenten

- White-Box Darstellung
- Stereotyp: <<subsystem>>



Artefakte

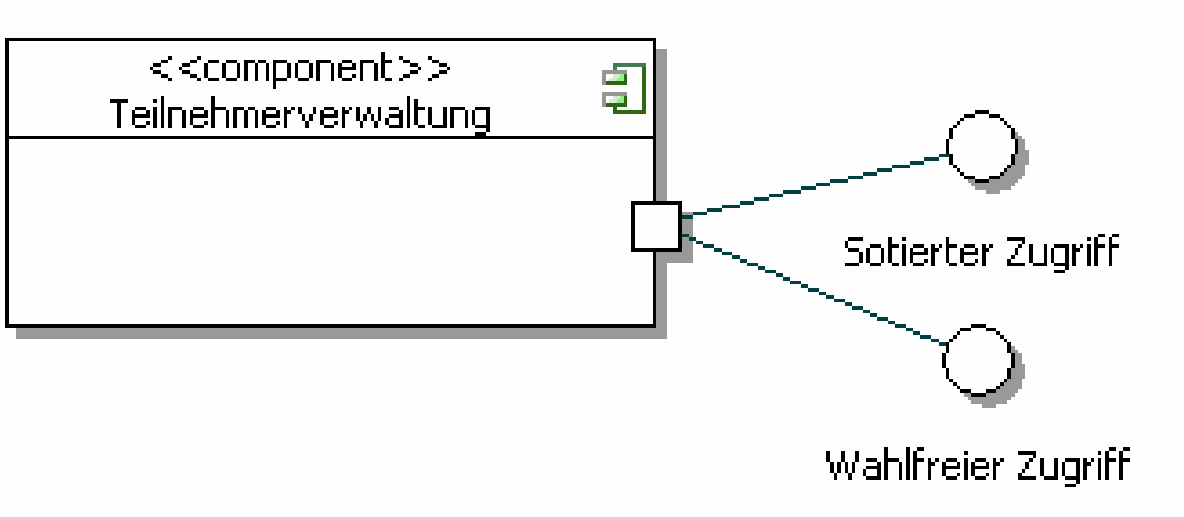
- konkretes Element in der realen Welt
- Komponente wird mit Artefakt manifestiert
- Stereotypen:



«file»	**Datei**
«document»	**weder Quell- noch Zieldatei**
«executable»	**ausführbare Datei**
«source»	**Quelldatei**
«libary»	**statische/dynamische Bibliothek**

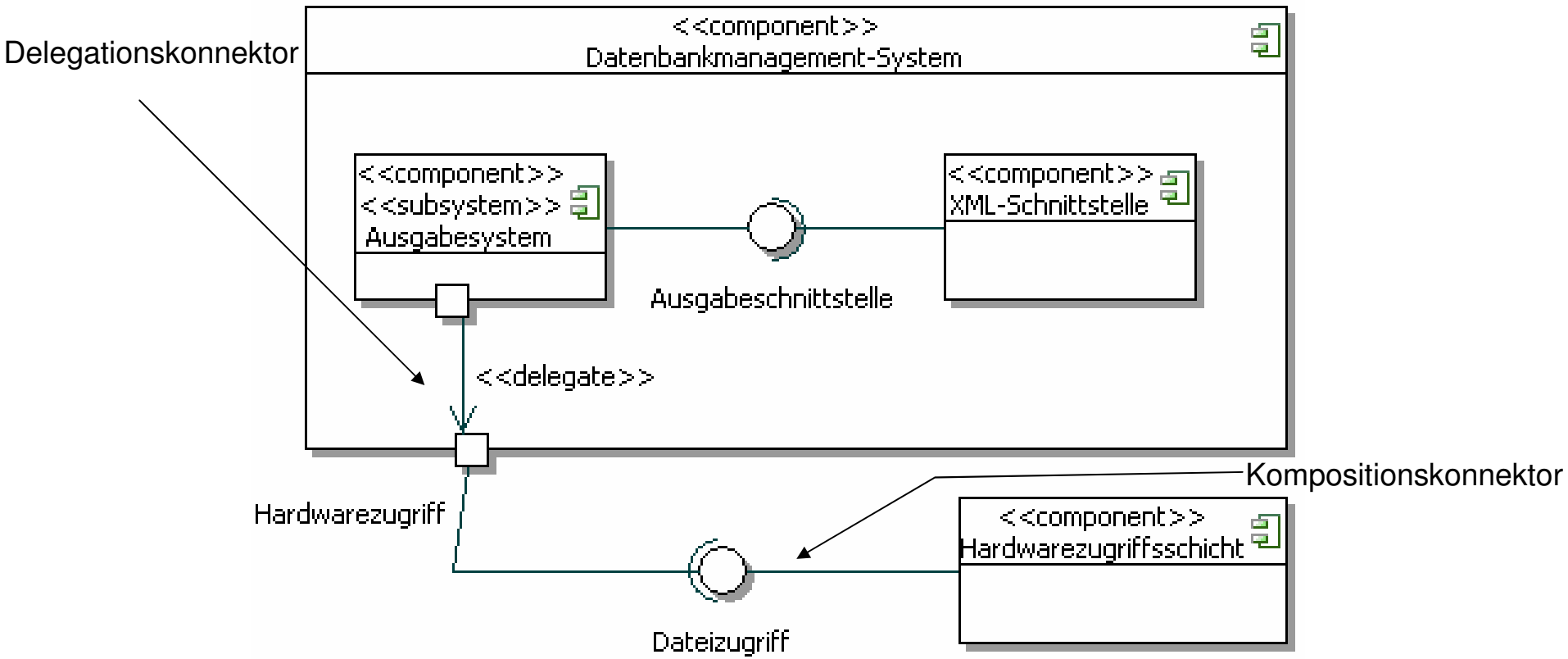
Ports

- Interaktionspunkt zwischen der Komponente und der Umgebung
- Funktionen direkt implementiert oder weitergeleitet.



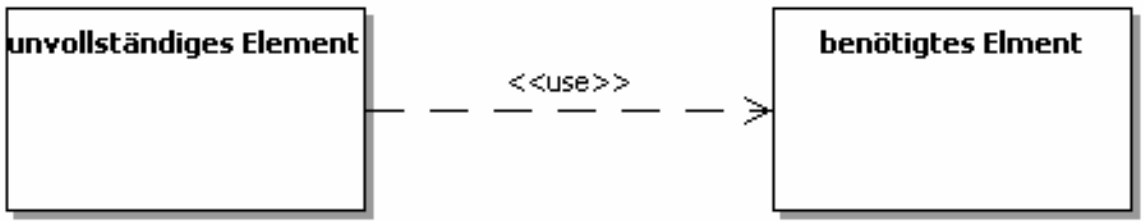
Konnektoren

- andere Bedeutung als in ADLs



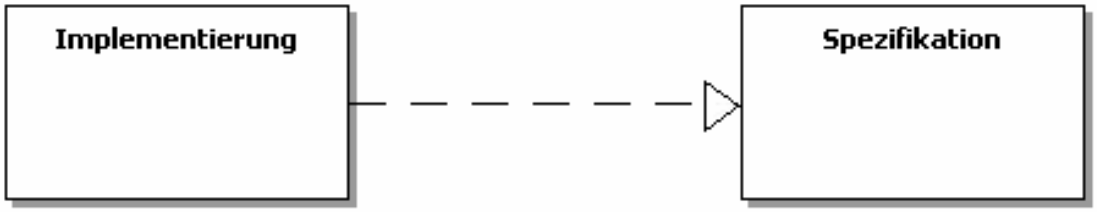
Beziehungen

- Verwendungsbeziehung
  - <<use>>



    - unvollst. Element verwendet benötigtes Element

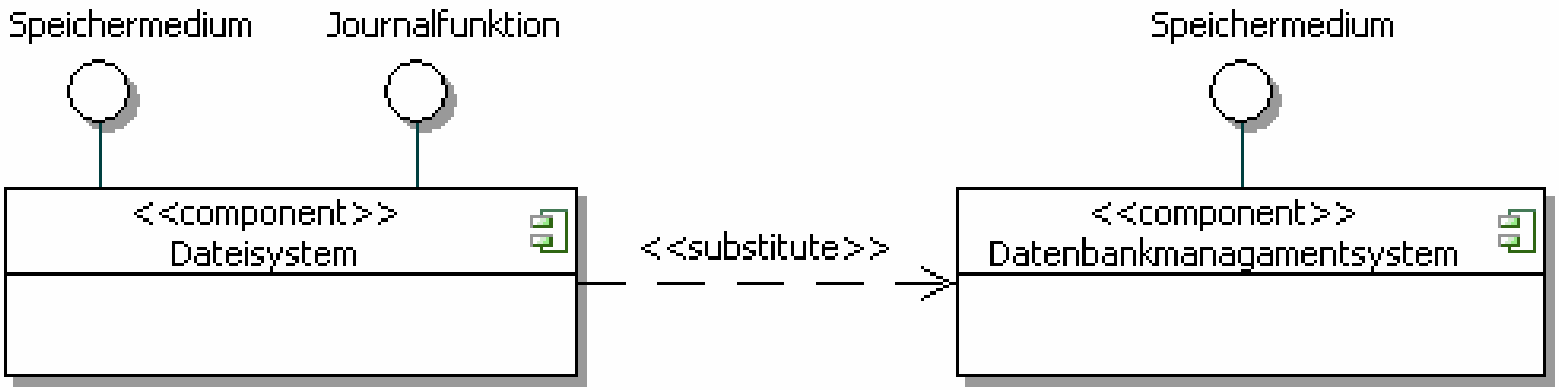
- Realisierungsbeziehung
  - <<realize>>



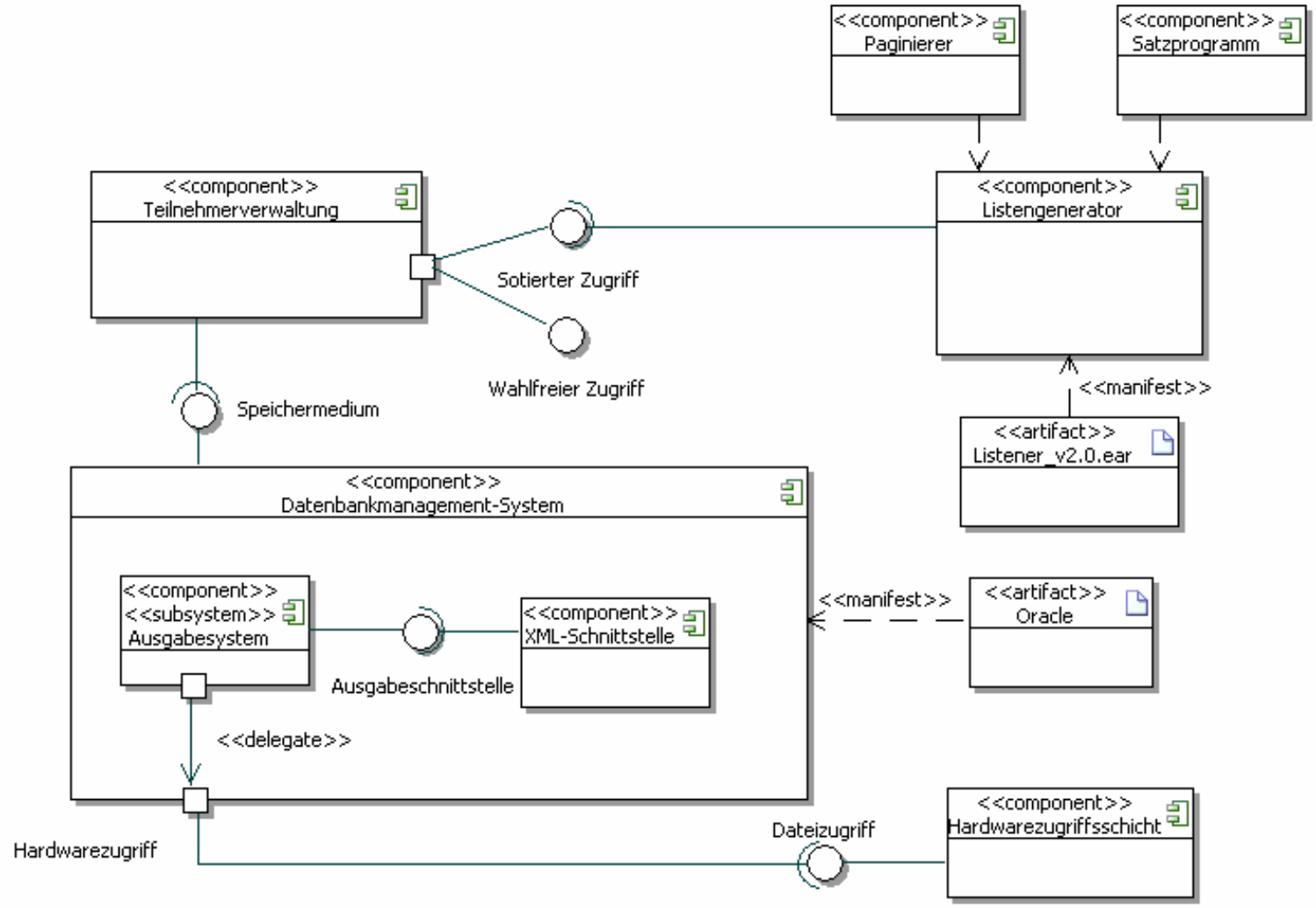
    - Spezifikation wird durch Implementierung realisiert

Beziehungen

- Substitutionsbeziehung
  - <<substitute>>
    - eine Komponente kann zur Laufzeit durch eine andere ersetzt werden



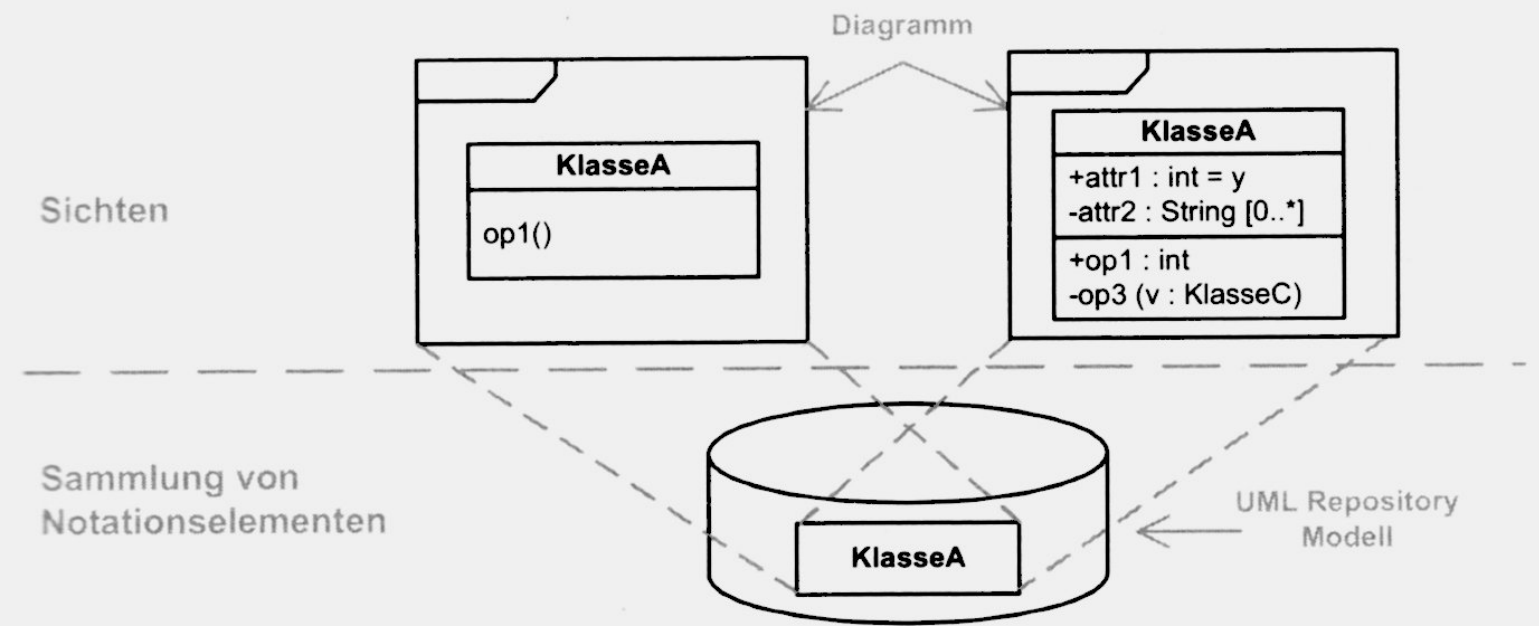
Architectural Configuration



weitere Diagramme / Sichten um UML als ADL zu nutzen

- Kompositionsstrukturdiagramme
  - Beschreibung der Feinstruktur von Komponenten
- Protokoll Zustandsautomat
  - Beschreibung von dynamischen Schnittstellen
- Sequenzdiagramme
  - Beschreibung des Verhaltens Komponenten

Sichten und Diagramme



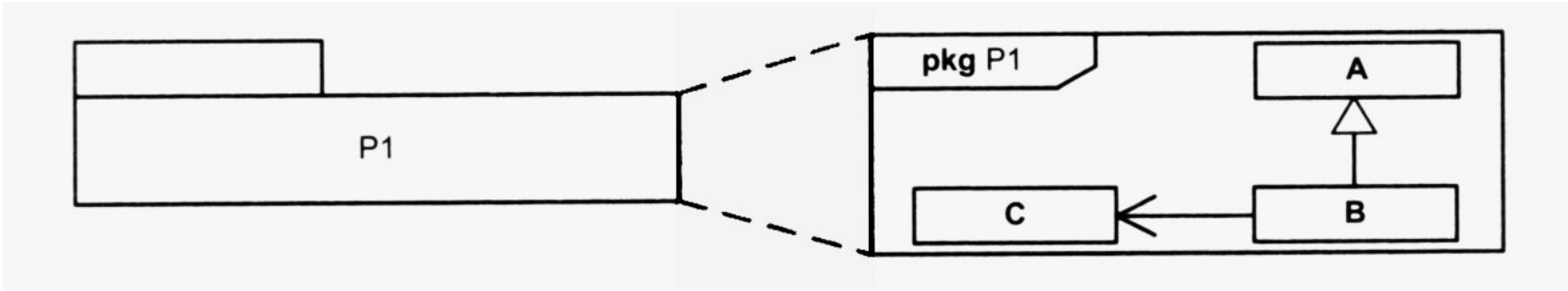
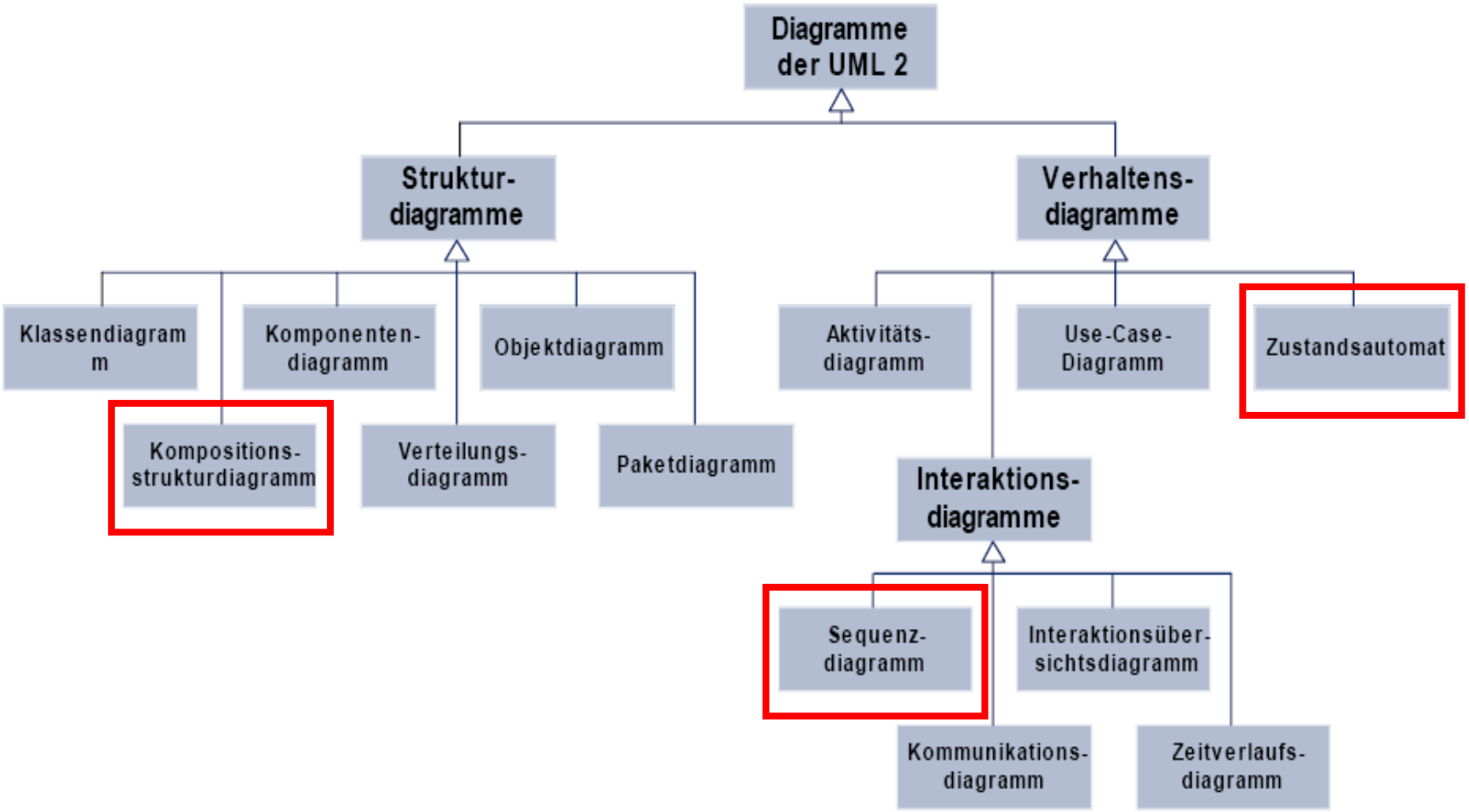
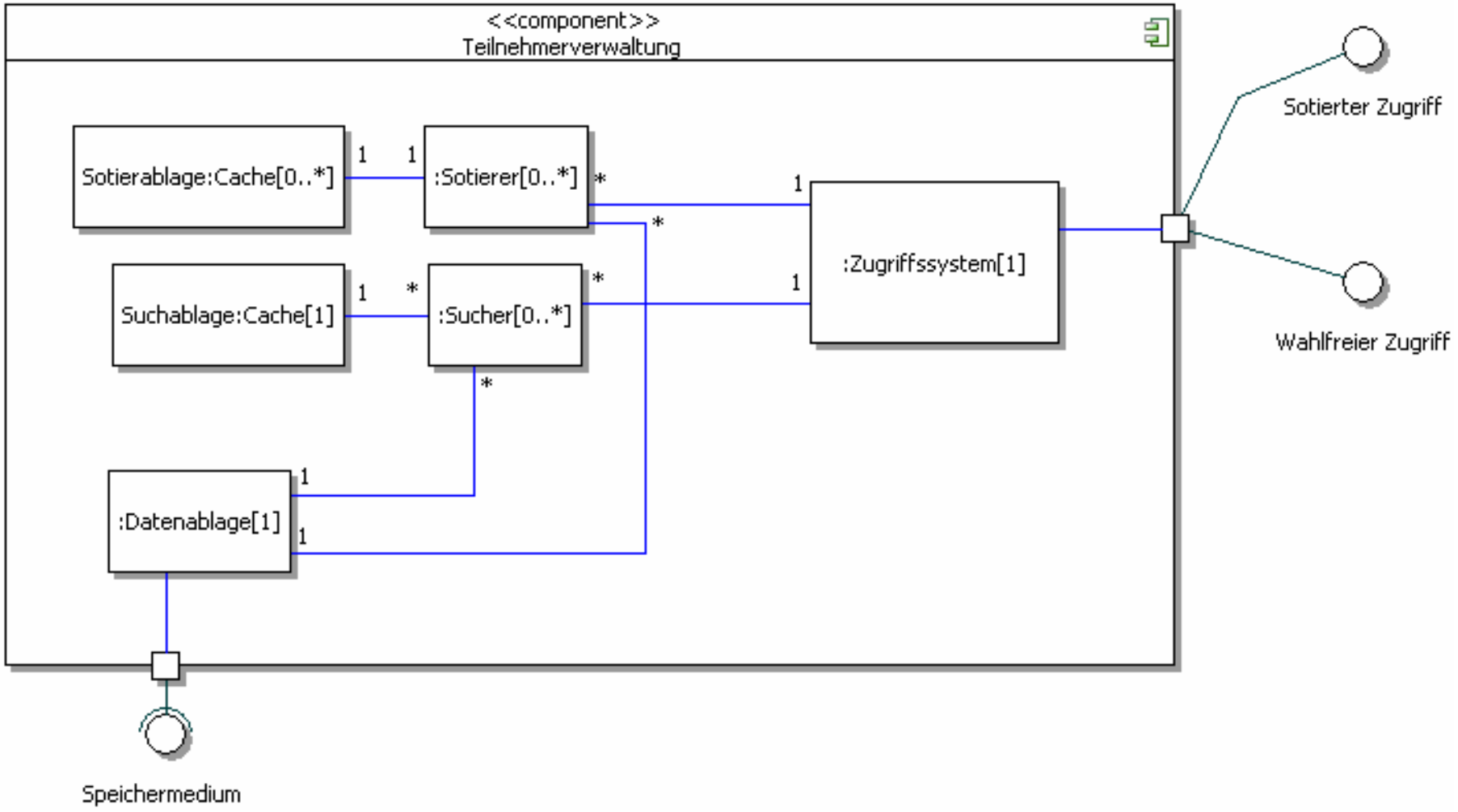


Diagramme der UML 2.0

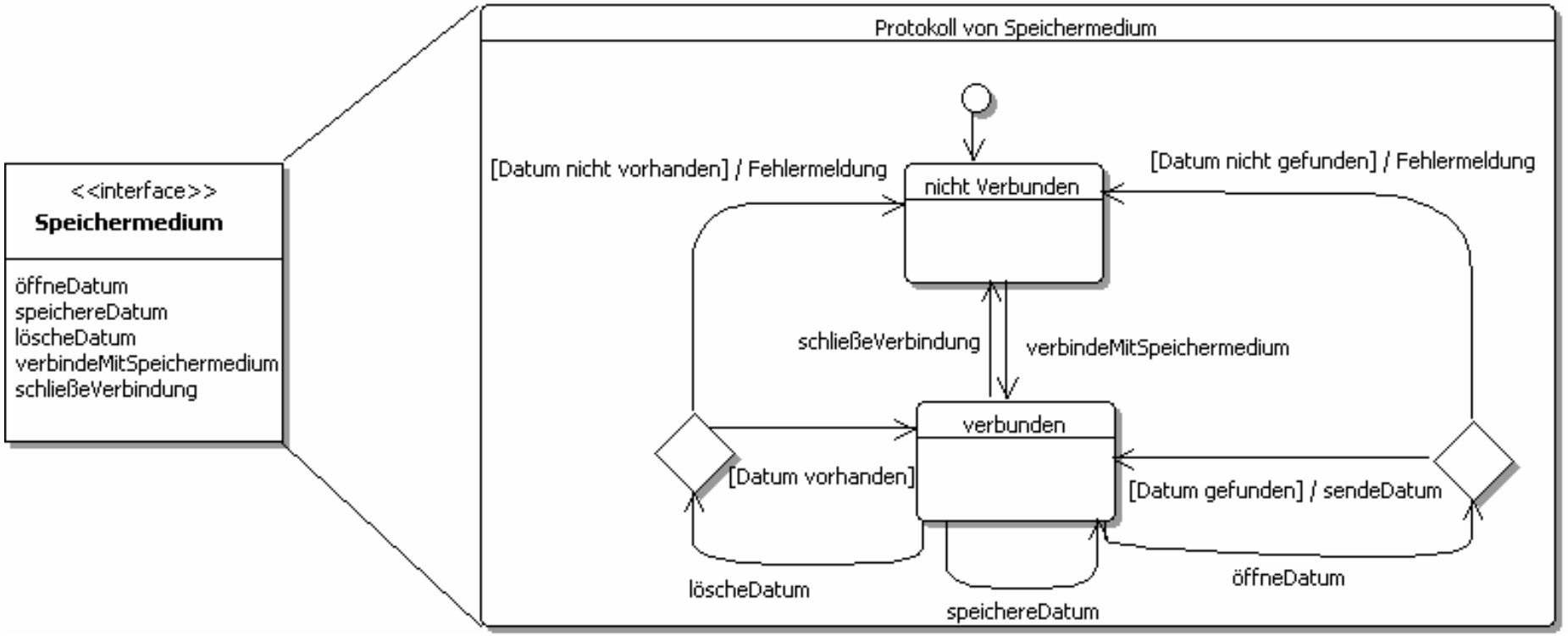


<<component>> Einleitung
<<component>> Komponentendiagramme
<<component>> weitere Diagramme
<<component>> Fazit

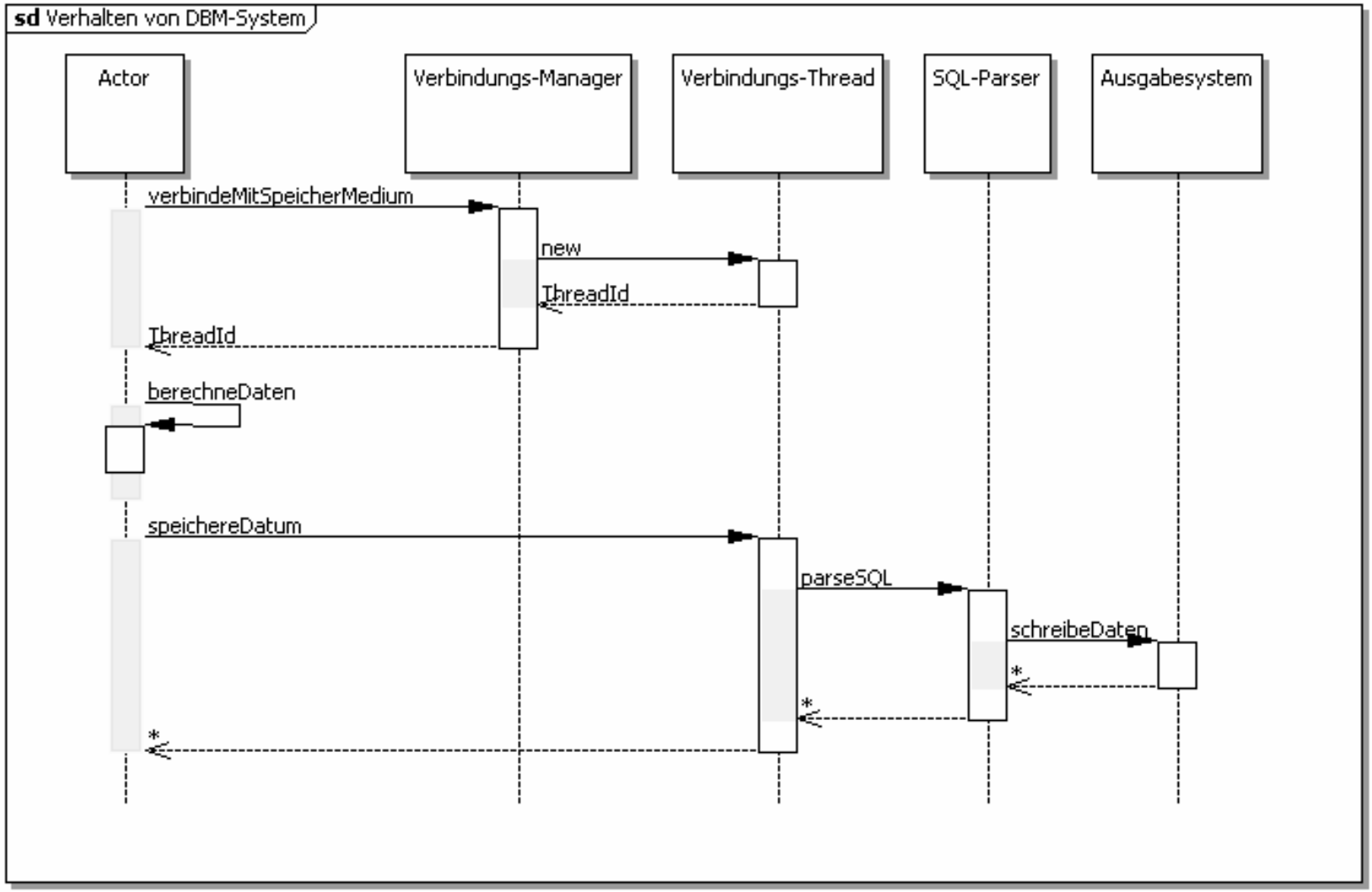
Kompositionsstrukturdiagramme



Protokoll Zustandsautomaten



Sequenzdiagramme



<<component>> Einleitung
<<component>> Komponentendiagramme
<<component>> weitere Diagramme
<<component>> Fazit

Ausblick

- Profil für die Architekturbeschreibung in UML
  - Syntax und Semantik einschränken
  - Werkzeugsystem zur Analyse und Simulation
  - vielleicht zukünftiges ADL-Profil was auf Komponentendiagrammen aufbaut?

Fazit

- Modellierung von Architektur in UML möglich (Ausnahme von Konnektoren)
- restliche Spezifikation in UML → Architektur in UML
- Profile
  - Semantik einschränken
  - Werkzeugsysteme müssten erstellt werden
- domainspezifische Probleme → domainspezifische ADL
- allgemeine Probleme →möglicherweise UML 2.0

Vielen Dank für Ihre Aufmerksamkeit

Fragen ?

Quellen

- UML 2.0 glasklar, Chris Rupp, Jürgen Hahn, … , 2. Auflage, 2005 Carl Hanser Verlag, ISBN: 3-446-22952-3,
- UML Superstructure, version 2.0, formal/05-07-04
- A Classification and Comparison Framework for Software Architecture Description Languages, Taylor and Medvidovic
- Wikipedia