



# Palladio Componentmodel

Entwurfsbeschreibung

Marko Hoyer

*Marko.Hoyer@informatik.uni-oldenburg.de*

24. September 2005

# Inhaltsverzeichnis

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Einleitung</b>   | <b>2</b> |
| <b>2</b> | <b>Architektur</b>  | <b>2</b> |
| <b>3</b> | <b>Datenhaltung im Modellkern</b>                             | <b>4</b> |
| <b>4</b> | <b>Instanziierung des Modells</b>                             | <b>4</b> |
| <b>5</b> | <b>Aufbau eines neuen Modells</b>                             | <b>4</b> |
| <b>6</b> | <b>Benachrichtigung bei Änderungen im Modell</b>              | <b>4</b> |
| <b>7</b> | <b>Suchanfragen an das Modell</b>                             | <b>4</b> |
| 7.1      | Allgemeine Anfragen . . . . .                                 | 4        |
| 7.2      | Navigation im Modell . . . . .                                | 4        |
| 7.3      | Vergleichbarkeit zwischen Bestandteilen des Modells . . . . . | 4        |
| <b>8</b> | <b>Persistente Speicherung des Modells</b>                    | <b>4</b> |
| <b>9</b> | <b>Erweiterungsmöglichkeiten</b>                              | <b>4</b> |
|          | <b>Literaturverzeichnis</b>                                   | <b>4</b> |

## 1 Einleitung

## 2 Architektur

In diesem Kapitel wird die derzeitige Architektur des Komponentenmodells vorgestellt. Sie setzt sich aus den in Abbildung 1 dargestellten und im Folgenden kurz erläuterten Bestandteilen zusammen. An der Umrandung der Blöcke ist abzulesen, ob diese bereits entworfen oder lediglich als Erweiterungen geplant sind. Details und Informationen zur Umsetzung der Bestandteile des Komponentenmodells sind Inhalt der folgenden Kapitel dieses Dokuments.

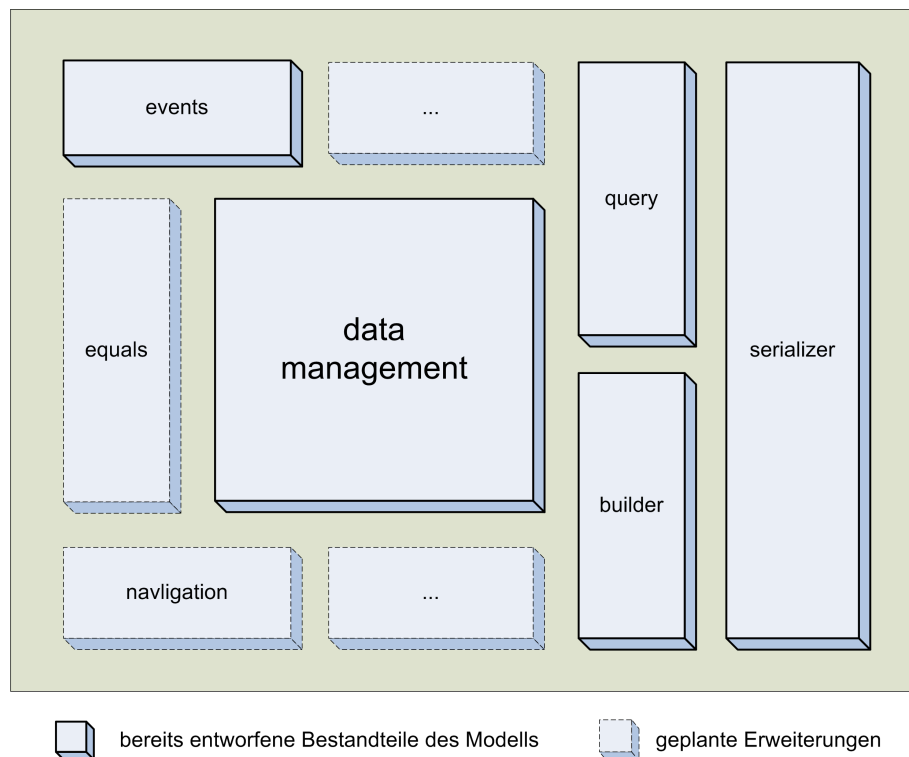


Abbildung 1: Architektur des Komponentenmodells

Zentrum der Architektur bildet die in der Abbildung mit *data management* bezeichnete Datenhaltung. Sie dient der lokalen Speicherung der Entitäten und Relationen des Modells zur Laufzeit der nutzenden Anwendung. Hierfür stellt sie Möglichkeiten zum Lesen und Schreiben der Daten zur Verfügung. Die Konsistenz der Daten ist in dieser Schicht lediglich in Bezug auf die verwendeten Datenstrukturen zu gewährleisten. Semantische Fehler im Sinne des theoretischen Komponentenmodells sind von der Datenhaltung zu tollerieren,

um unabhängig von möglichen Änderungen dieser Semantik zu bleiben. Aufgrund dessen ist der das Modell nutzenden Anwendung keine Möglichkeit zu gewähren, direkt auf die Datenhaltung zuzugreifen, da sonst Korrektheit im Sinne des theoretischen Modells nicht mehr gewährleistet werden kann. Zugriff ist erst nach Überprüfung durch entsprechende Zwischenschichten zu gestatten.

Schreibende Änderungen der Anwendung am Modell sind hierbei durch den in der Architektur mit *builder* bezeichneten Block vorgesehen. Dieser stellt eine Infrastruktur bereit, welche Änderungen an verschiedenen Stellen des Modells zulässt und den korrekten Aufbau gemäß dem theoretischen Modell sicherstellt. Es können an dieser Stelle bereits durch geschickte Wahl der Zugriffsmethoden Fehler ausgeschlossen werden. Eine Möglichkeit der Umsetzung dieser Schicht unter Beachtung der Hierarchie des Komponentmodells wird in Kapitel 5 vorgestellt.

Der lesende Zugriff auf das Modell kann je nach Bedarf durch verschiedene Schichten erfolgen. Die in Abbildung 1 mit *query* bezeichnete Schicht dient der Abfrage von Attributen der Entitäten und der Beziehung zwischen diesen. Eine direkte Abfragemöglichkeit der Datenhaltung ist prinzipiell möglich, jedoch aufgrund fehlenden Wissens über das theoretische Modell unpraktikabel. Abstraktionen in diesem Sinn sind also ebenfalls Aufgabe der Abfrageschicht.

- 3 Datenhaltung im Modellkern
- 4 Instanziierung des Modells
- 5 Aufbau eines neuen Modells
- 6 Benachrichtigung bei Änderungen im Modell
- 7 Suchanfragen an das Modell
  - 7.1 Allgemeine Anfragen
  - 7.2 Navigation im Modell
  - 7.3 Vergleichbarkeit zwischen Bestandteilen des Modells
- 8 Persistente Speicherung des Modells
- 9 Erweiterungsmöglichkeiten

Literatur