

Verteidigung Diplomarbeit

Vergleichende Studie zu den
Komponentenmodellen COM+, EJB und CCM

Stefan Haubold
(Stefan.Haubold@web.de)

Gliederung

Aufgabenstellung

Grundlagen

Problem/Motivation

Vorgehensweise

Modellierung mit den
Komponentenmodellen

Zusammenfassung/Ausblick

Literaturhinweis

Aufgabenstellung

Einarbeitung in die Komponentenmodelle

Literaturrecherche

Vergleich der Komponentenmodelle

Vergleich anhand eines Beispiels (UML)

Schlußfolgerungen

Grundlagen (1)

Komponente

unabhängiger, austauschbarer Softwarebaustein

Schnittstelle

Menge von Operationen, Vertrag

Komponentenmodell

Rahmen für Komponenten, Regeln, Standard

EJB, CCM, COM+

Container, Dienste

Grundlagen (2)

Cheesman/Daniels „UML Components“

komponentenmodellunabhängig

Schnittstellenspezifikationen

Komponentenspezifikationen

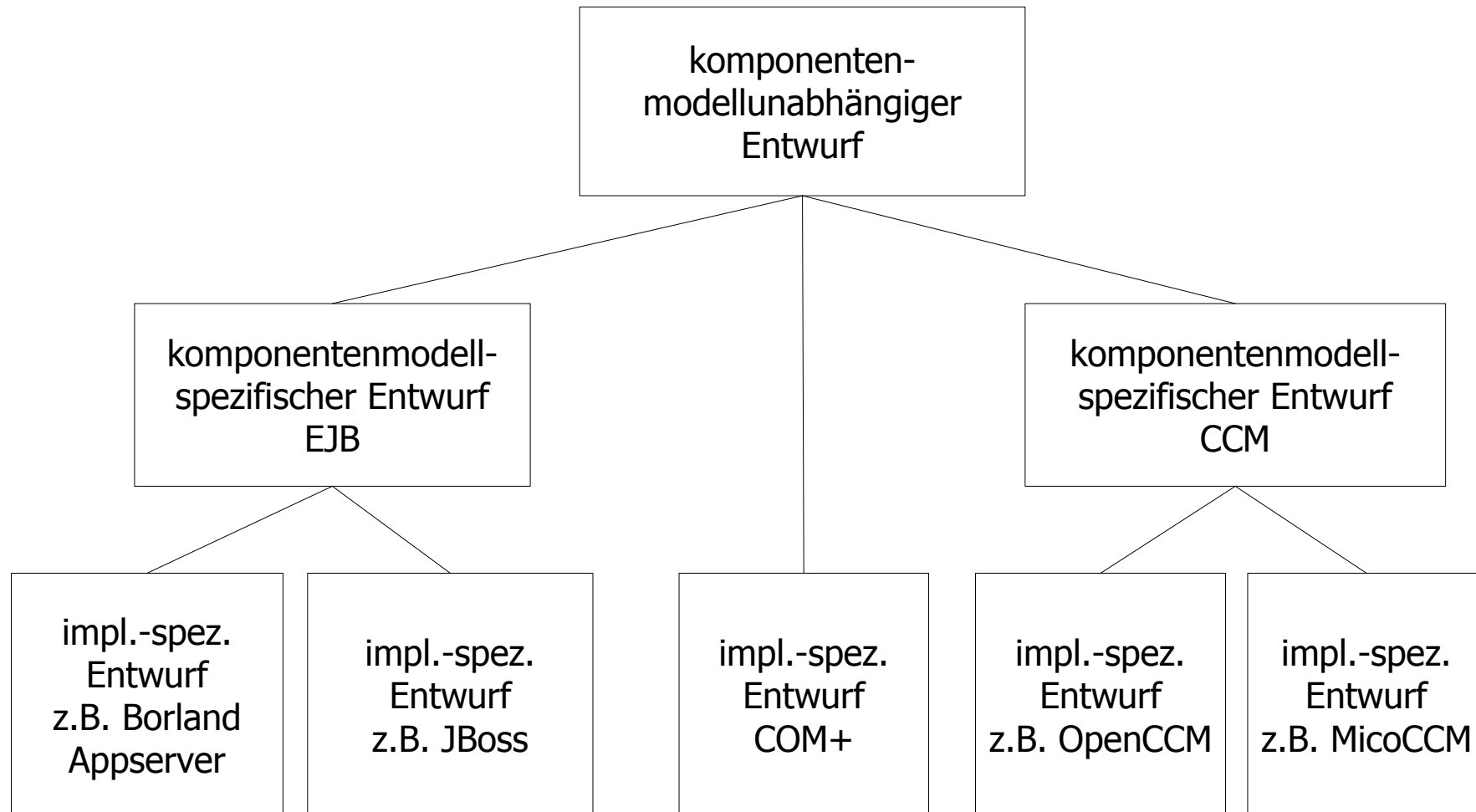
- Systemkomponenten

- Business-Komponenten

Das Problem

Wie kann ein komponentenmodellunabhängiger Entwurf auf ein konkretes Komponentenmodell abgebildet werden?

Detailierungsgrade beim Entwurf



Motivation

Wiederverwendung von Entwürfen

Portierung

einer Anwendung

einer Komponente

Abbildungsvorschriften

Automatisierung

Generator erzeugt Abbildung auf Komponentenmodell

Vorgehensweise

Realisierung von

Schnittstellenspezifikation

Komponentenspezifikation

Systemkomponenten

Business-Komponenten

Persistenz

Modellierung komponentenbasierter Anwendungen mit EJB

Java-Schnittstellen, Java-Klassen

nur eine fachliche Schnittstelle pro
Komponente

Zusammenfassen oder Aufteilen

feingranulare Entity-Beans

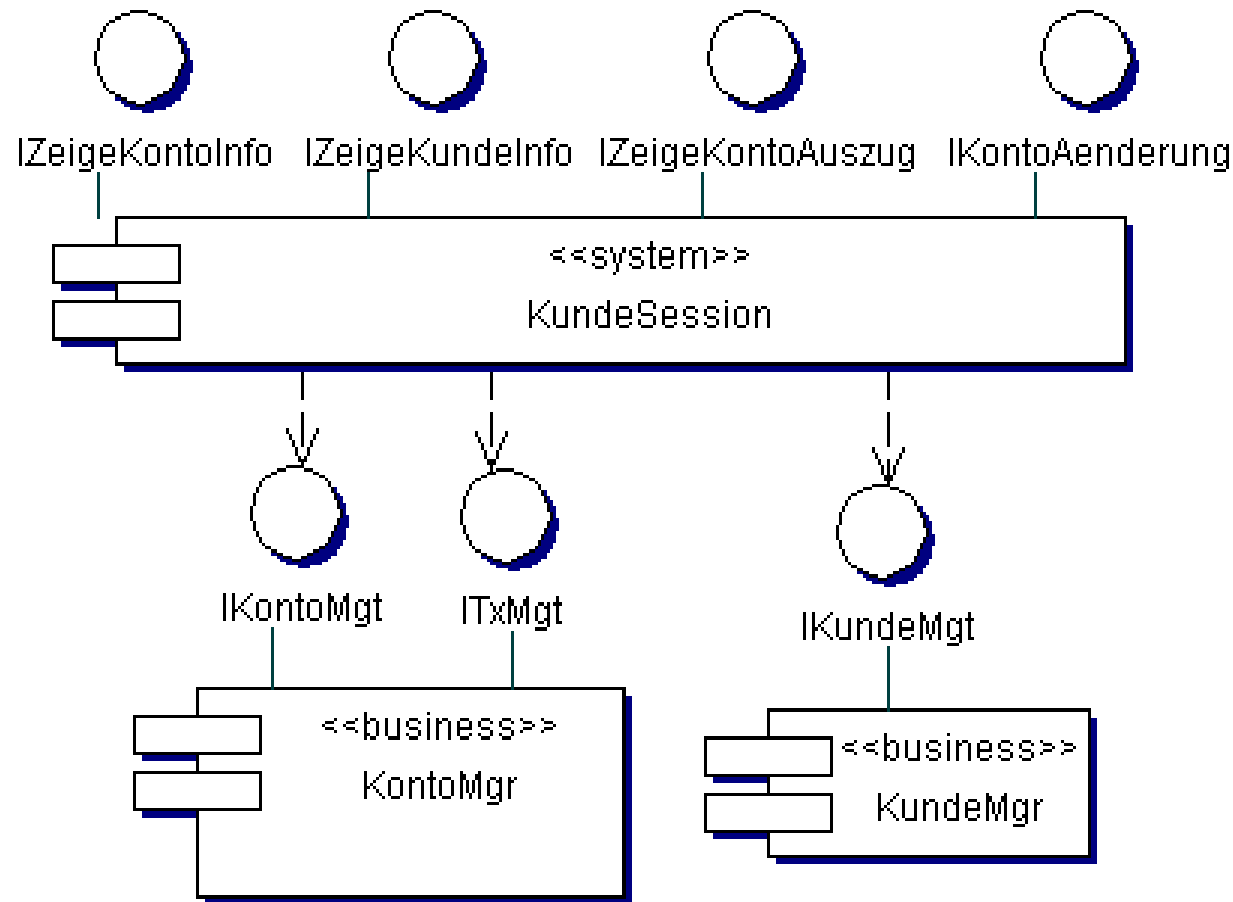
Komponentenbausatz, Unterkomponenten

Session-Fassade zur Koordination

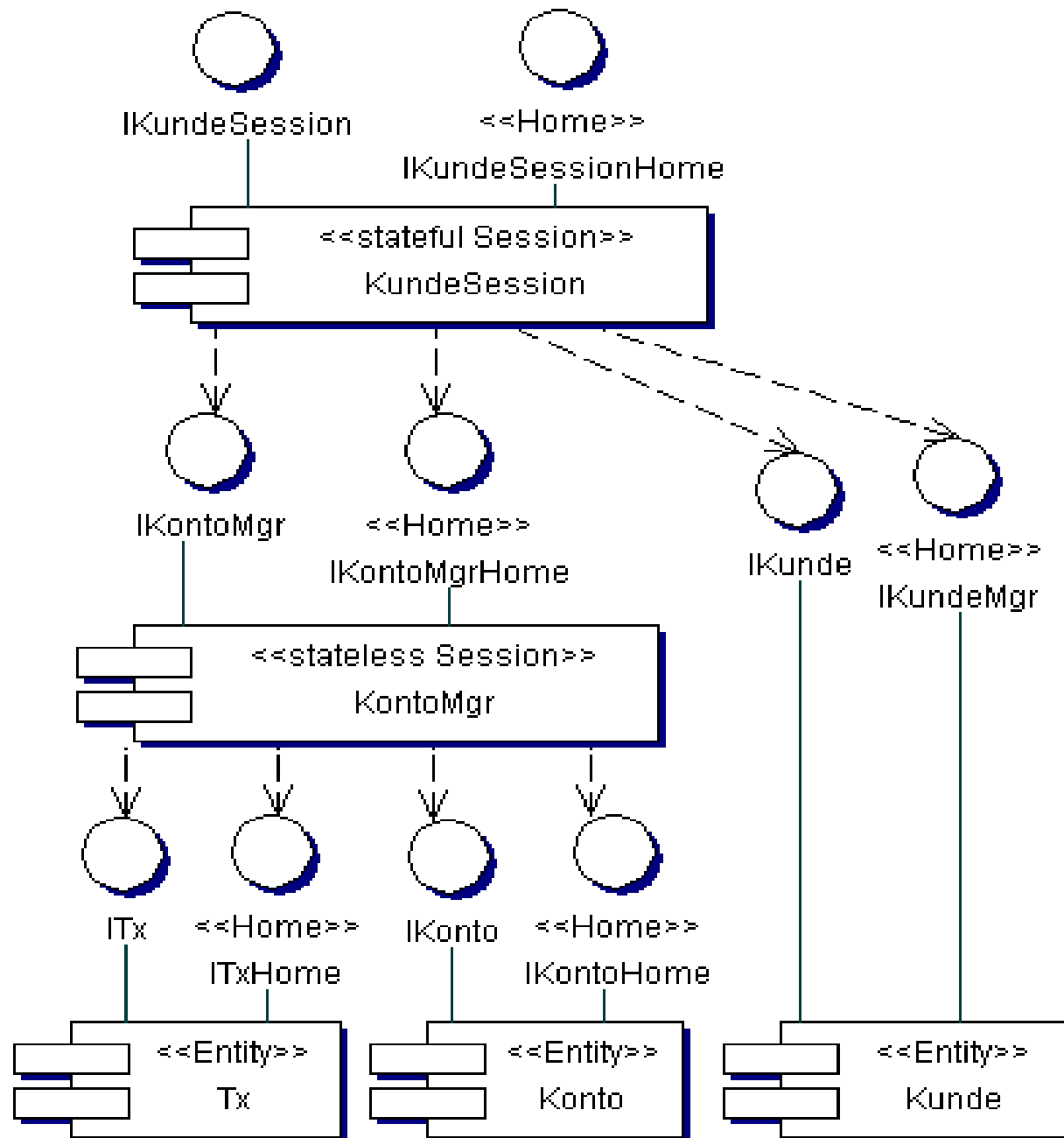
Entity-Beans mit lokalen Schnittstellen

Session-B., Entity-B. BMP, CMP, DAO

Komponentenmodellunabhängiger Entwurf



Komponentenmodellspezifischer Entwurf EJB



Modellierung komponentenbasierter Anwendungen mit CCM

baut auf EJB 1.1 auf

zusätzlich Events, Process-Komponenten

IDL:

Schnittstellen, Komponenten, Homes, Exceptions, Events

Abbildung auf Progr.-spr. durch IDL-Compiler

mehrere Schnittstellen pro Komponente

auch hier Entity-Komponenten

Komponentenbausatz, Unterkomponenten, keine lok.

Entity-K. BMP, CMP, Process-K.

Modellierung komponentenbasierter Anwendungen mit COM+

COM-Schnittstellen, COM-Klassen

auch hier (Microsoft-)IDL

MIDL-Compiler oder integriertes Tool der IDE

mehrere Schnittstellen pro COM-Klasse

Automation: nur eine Default-Schnittstelle!

Zusammenfassen oder Aufteilen

keine persistenten Komponenten

keine automatische Persistenz

Handarbeit mit ADO, OLE DB

Ergebnisse

überall: Schnittstellen + Komponenten

Granularitätsunterschiede erfordern zusammengesetzte Komponenten

universale Abbildungsvorschriften nicht erkennbar

Abstraktion -> Informationsverlust

Nützlichkeit eines komponentenmodellunabhängigen Entwurfs fraglich

Zusammenfassung

Architekturen ähnlich (besonders EJB + CCM)

trotzdem Unterschiede

Komponentenmodell ist mehr als nur
Parameter

Skalierbarkeit, Performance

- nicht global regelbar

- muss in Entwurf einfließen

verschiedene Philosophien

- 2 Lager: EJB und Microsoft

Ausblick

EJB 2.0 seit September

Umstellung der Implementierungen

EJB-Assembly-Diagramme

CCM Finalisierung (30.11.2001)

keine vollständige Referenzimplementierung

kommerzieller Erfolg ungewiß

COM+ geht in .NET auf

UML bald in Version 2.0

soll Komponentenmodellierung besser unterstützen

Literaturhinweis

Otto Kruse. Keine Angst vor dem leeren Blatt.
Ohne Schreibblockaden durchs Studium.
Campus Verlag, 2000.

Grundlagen des Schreibens

Schreibprobleme

wissenschaftliches Schreiben

Erkenntniswege, Textmuster

Arbeitsschritte

Zeitplanung

Fragen?