Zwischenbericht Diplomarbeit

Vergleichende Studie zu den Komponentenmodellen COM+, EJB und CCM

Stefan Haubold
(sh6@inf.tu-dresden.de)

Gliederung

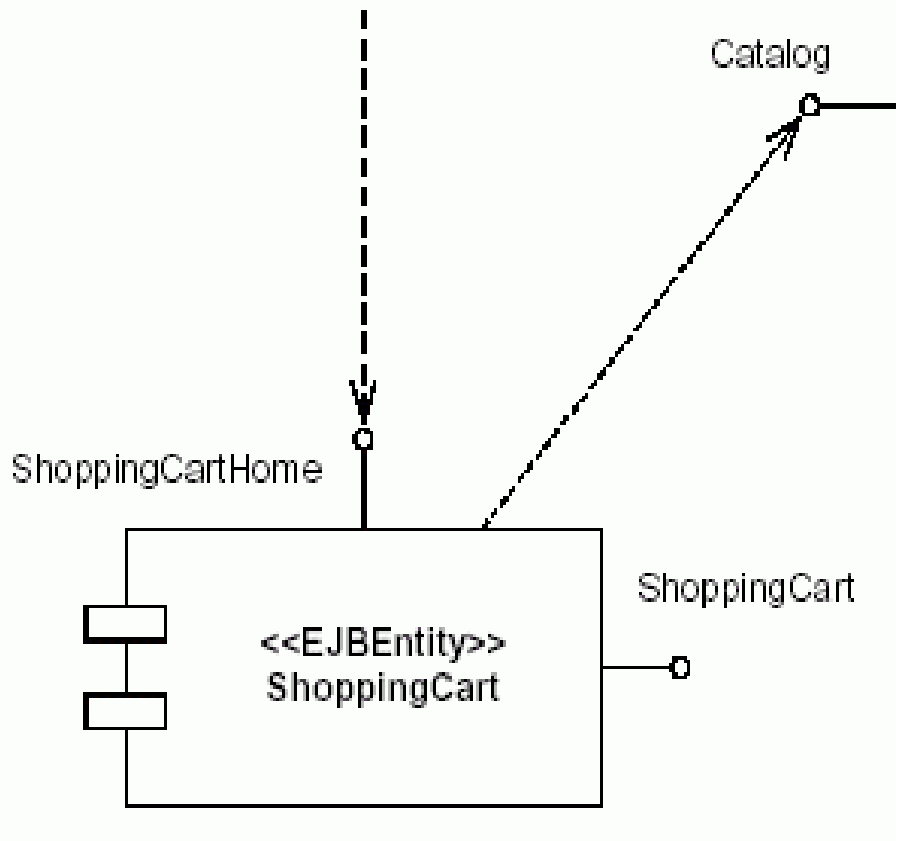
- Aufgabenstellung
- COM+
- Enterprise JavaBeans (EJB)
- CORBA Component Model (CCM)
- Beispiel
- Modellierung mit UML
- Architekturvergleich
- Abbildungsmöglichkeiten
- Zusammenfassung
- TODO

Aufgabenstellung

- Einarbeitung in die Komponentenmodelle
- Literaturrecherche
- Allgemeiner Vergleich, Architekturvergleich
- Vergleich anhand eines Beispiels (UML)
- Schlußfolgerungen

Komponente

- austauschbarer Softwarebaustein
- Trennung von Schnittstelle und Implementierung
- Schnittstelle ist Vertrag
- kapselt Funktionalität



Komponentenmodell

nach Gruhn, Thiel „Komponentenmodelle“:

- Rahmen für Entwicklung und Ausführung von Komponenten
- legt Verknüpfungs- u. Kompositionsmöglichkeiten sowie Kollaborationsmöglichkeiten fest
- Infrastruktur für Dienste wie Verteilung, Persistenz, Nachrichtenaustausch, Sicherheit und Versionierung

COM+

- Teil von Windows Distributed InterNet Architecture (Windows DNA), Vorläufer von .NET
- (D)COM + MTS + MSMQ + Erweiterungen = COM+
- MTS = Microsoft Transaction Server
- OTM und Application-Server
- MSMQ -> Queued Components
- Win2K = Referenzimplementierung von COM+ 1.0
- In-Memory-DB und Load Balancing nicht in Final von Win2K (Load Balancing: AppCenter Server)

Enterprise JavaBeans

- Teil von J2EE (mit JMS, JTA, JavaMail, JSP, ...)
- aktuell Spezifikation 1.1, Vorabversion (Public Draft 2) von 2.0
- J2SDK EE 1.3 Beta 2 Referenzimplementierung
- viele kommerzielle EJB-Server (BEA, Borland, Sybase, IONA, Oracle, IBM)
- Session- und Entity-Beans

EJB 2.0 – What's new?

- Message-Driven-Beans
- Lokale Schnittstellen
- Container Managed Persistence überarbeitet
- Container Managed Relationships
- EJB Query Language
- Sicherheit: run-as ...
- instanzunabhängige Home-Methoden

CORBA Component Model

- Teil von CORBA 3, Spezifikation an EJB angelehnt, programmiersprachenunabhängig
- Session- und Entity-Komponenten plus neuer Typ: Prozess-Komponenten

Komponenten-kategorie	Container-typ	Containerim-plementierungstyp	Primär-schlüssel	EJB-Pendant
Service	transient	zustandslos	nein	stateless Session Bean
Session	transient	zustandsbehaftet	nein	stateful Session Bean
Process	persistent	durable	nein	-
Entity	persistent	durable	ja	Entity Bean

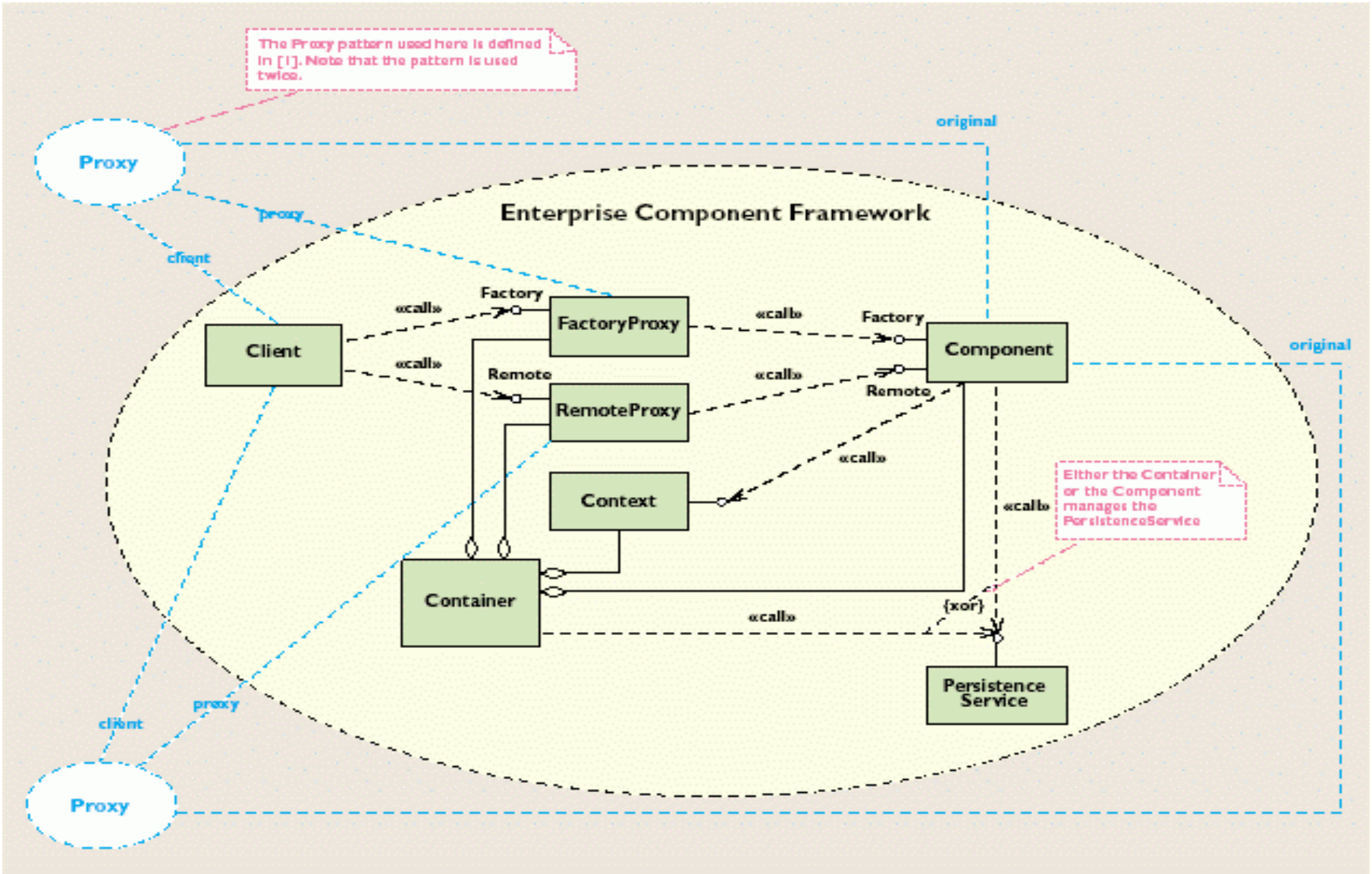
Beispiel

- Duke’s Bank aus J2EE-Tutorial
- Kunden, Konten, Geld einzahlen, abheben, überweisen usw.
- Übertragung auf CCM relativ einfach, Problem: Implementierung noch Alpha
- Übertragung auf COM+ schwieriger, da kein EB-Äquivalent und zustandslose Komponenten

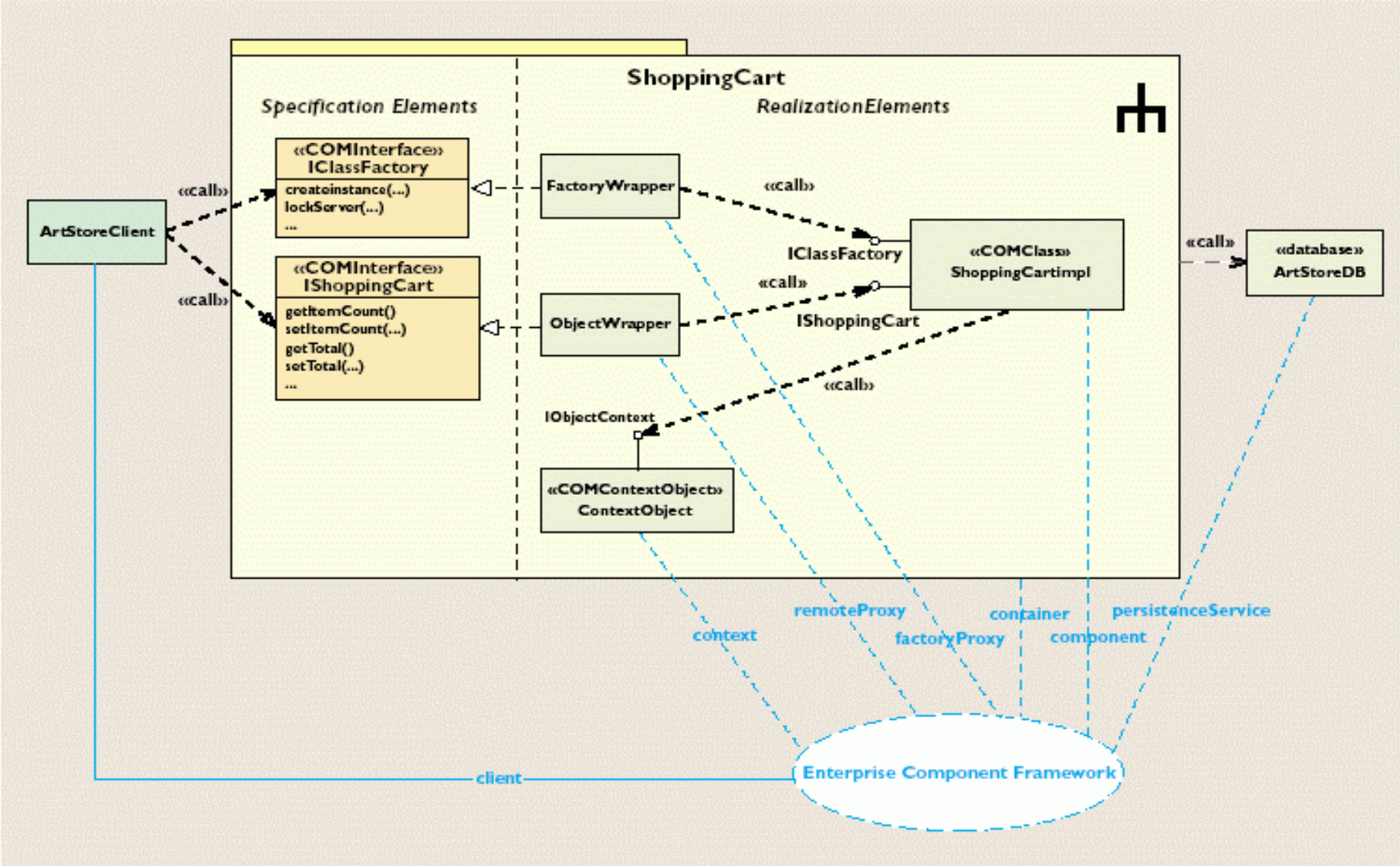
Komponentenmodellierung mit UML

- UML 1.3: Komponenten als Artefakte der Impl.-phase, aber mit Subsystemen nachzubilden
- UML 1.4: Komponentendiagramm und Komponentenbegriff revidiert, auch Entwurf
- UML Profile for EJB
- Erweiterung mit Stereotypen

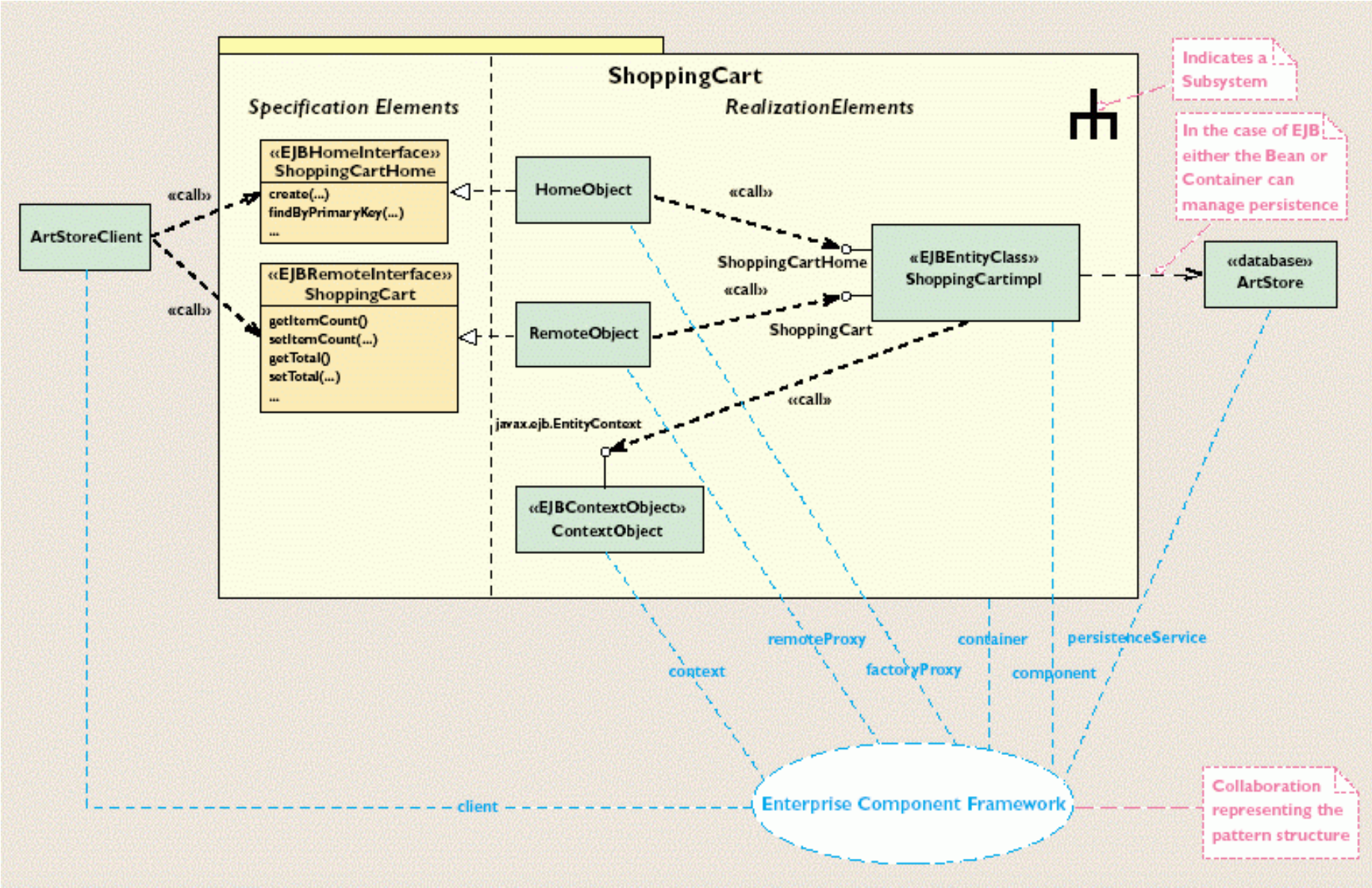
Container-Architekturmuster



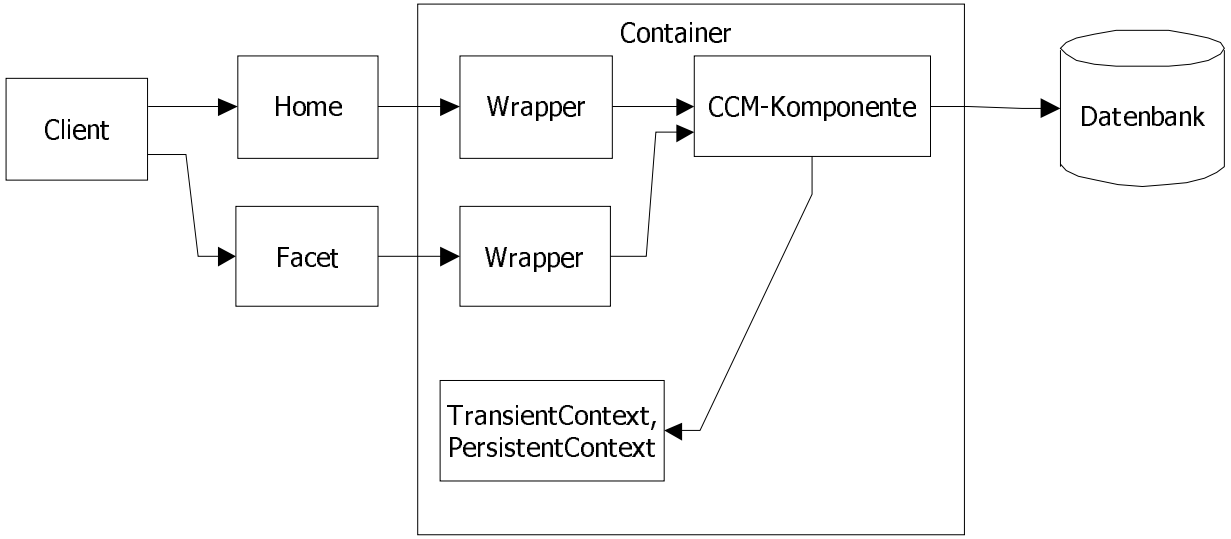
Container-Muster mit COM+



Container-Muster mit EJB



Container-Muster mit CCM



Abbildungsmöglichkeiten

- Komplexität: COM+ am einfachsten, EJB mittel, CCM am komplexesten
- Abbildung COM+ -> EJB -> CCM relativ einfach, umgekehrt schwierig, da Komplexitätsreduktion nötig
- z.B. COM+-Komp. -> stateless SessionBean -> Service-Komp.
- Message-Bean -> Queued Component
- kein Entity-Äquivalent bei COM+ -> Handarbeit, aber: ADO, OLE DB, ODBC

Zusammenfassung

	CCM	EJB	COM+
Architekturkonzept	Container & Interception	Container & Interception	Container & Interception
Muster zum Finden,Erzeugen	Finder & Factory	Finder & Factory	Factory
Schnittstellen pro Komponente	mehrere	eine	mehrere
Komponentenarten	Service, Session, Process, Entity	Session, Entity, Message	stateless Session, Queued
Kommunikationsprotokoll	IIOP	RMI-IIOP	DCOM
Container Managed Persistence	ja	ja	nein
Ereignisse	Notification Service	nur über JMS	COM+ Events
Sicherheit	Security Service	rollenbasiert & prog.	rollenbasiert & prog.
Programmiersprachen	diverse	Java	diverse
Plattform	diverse	Java	Windows 2000
Implementierungen	2 (Alpha)	viele	eine

(Zwischen-)Fazit

- Architekturen ähnlich (besonders EJB und CCM)
- auch große Unterschiede
- Abbildungsvorschriften möglich, aber:
  Komponentenmodell ist mehr als nur Parameter
- Skalierbarkeit, Performance nicht global regelbar
- verschiedene Philosophien

TODO

- Beispiele mit UML ausarbeiten (genaue Analyse und Entwurf) und implementieren
- Architekturvergleich fertigstellen
- Schlußfolgerungen für den Entwurf von komponentenbasierten Anwendungen

Fragen?