JAVAFORUM stuttgart 2001

plenum
the architect of new business

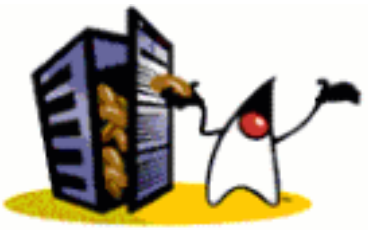
Enterprise Java Beans 2.0
Nutzen der neuen Spezifikation

Dipl. Inf. Volker Koch
Volker.Koch@plenum.de

plenum
the architect of new business

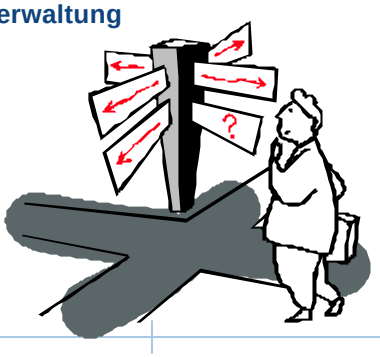
Agenda

- **Probleme der ersten EJB-Spezifikationen**
- **MessageDrivenBeans**
- **Der Persistence Manager Provider**
  - CMP Entity Beans
  - Dependent Objects
  - Objektbeziehungen
- **Die EJB Query Language**
  - Abfragen mit Finder-Methoden
  - Abfragen mit Select-Methoden
- **Die Erweiterung des Home-Interfaces**
- **Weitere Änderungen im Überblick**
- **Verfügbare Implementierungen von EJB 2.0**
- **Bewertung von EJB 2.0**



Probleme der ersten Spezifikationen

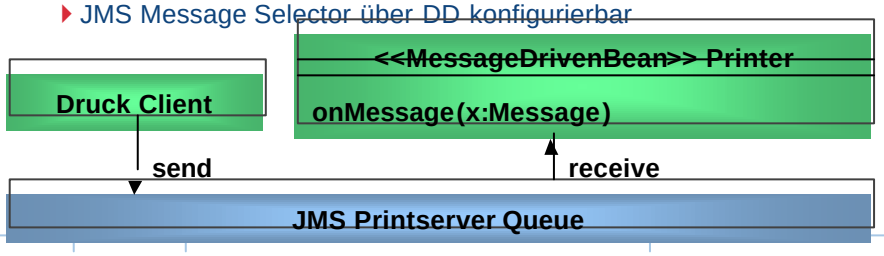
- **Tendenz Entity Beans zu feingranular zu entwerfen**
- **Mit CMP eingeschränkte Möglichkeiten Entity Beans auf Datenbank zu mappen (meist nur 1:1 Mapping)**
- **Keine Möglichkeit EJBs asynchron aufzurufen**
- **Keine automatische Beziehungsverwaltung**
- **Keine Vererbung zwischen EJBs**
- **Keine einheitlich Abfragesprache für Finder-Methoden**
- **Home Interface nicht erweiterbar**



3

MessageDrivenBeans

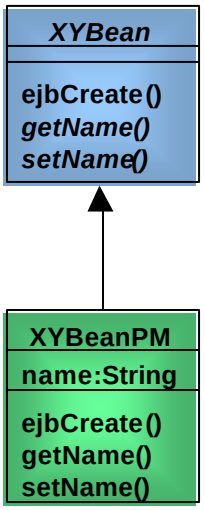
- **Neue, dritte EJB-Art**
  - Asynchroner und paralleler Empfang von Nachrichten über den Java Messaging Service (JMS)
  - Sowohl für JMS Queues, als auch für JMS Topics einsetzbar
  - Bean-Klasse ist ein JMS Message Listener
  - Besitzt weder Home- noch Remote-Interface
  - Kennt wie Stateless Session Bean keinen Status
  - JMS Message Selector über DD konfigurierbar



4

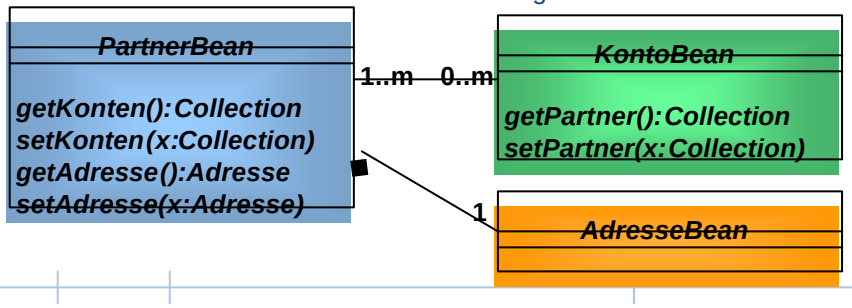
Der Persistence Manager Provider

- **7. Rolle im Entwicklungs- und Deployment-Prozeß**
- **Zuständig für die Persistenz von CMP Entity Beans**
- **Bean Provider schreibt abstrakte Bean-Klasse**
  - Enthält Business Logik
  - Abstrakte getter/setter-Methoden
  - keine persistenten Attribute
- **Persistence Manager Provider (PMP) erzeugt konkrete Unterklasse**
  - Anbindung an Datenhaltungssystem
  - Implementierung der getter/setter-Methoden
  - Überschreiben der Lebenszyklusmethoden (ejbCreate,…)



Der Persistence Manager Provider (2)

- **PMP zuständig für Beziehungsverwaltung**
  - Bean Provider schreibt abstrakte getter/setter-Methoden (nicht im Remote-Interface)
  - Persistence Manager erzeugt Implementierungen dazu
  - 1:1, 1:n, m:1, m:n Beziehungen
  - bi- oder unidirektionale Beziehungen
  - Kaskadierendes Löschen über DD konfigurierbar



Der Persistence Manager Provider (3)

EJB 1.1 CMP

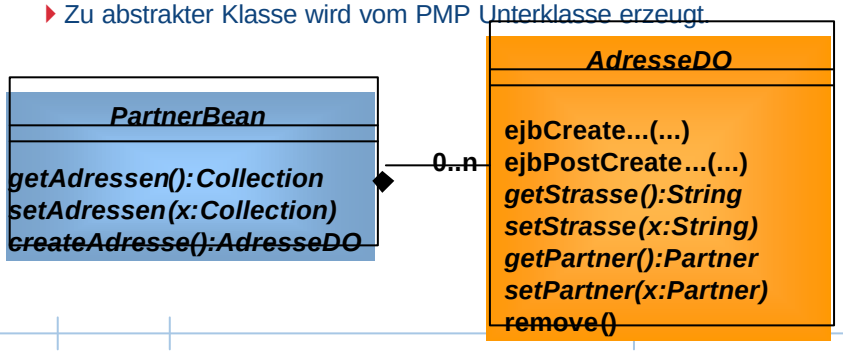
```
public class Bean ...{
public String name;
public Vector items;
public String getName(){
       return name;
}
public void setName(
       String pName){
       name = pName;
}
public void addItem(
       Item pItem){
       items.add(pItem);
}
...
}
```

EJB 2.0 CMP

```
public abstract class Bean ...{

public abstract String getName();

public abstract void setName(
       String pName);

public void addItem(Item pItem){
       getItems().add(pItem);
}
public abstract
       Collection getItems();
...}
```


Dependent Objects

- **Dependent Object (DO) ist Java Objekt**
- **Persistenz**
  - Besitzt Primärschlüssel
  - Keine persistenten Attribute
  - Abstrakte getter/setter Methoden
  - Zu abstrakter Klasse wird vom PMP Unterklasse erzeugt



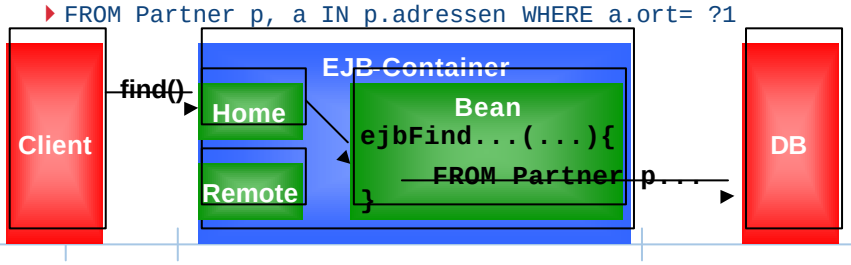
Dependent Objects (2)

- **Lebenszyklus wird von einem Entity Bean verwaltet.**
  - Bean-Klasse enthält abstrakte create<XXX> Methode.
  - DO-Klasse enthält ejbCreate<XXX>, ejbPostCreate<XXX>, remove
- **Beziehungen**
  - DO kann Beziehungen zu anderen DO oder Entity Beans haben.
- **Für den Client nicht sichtbar**
  - Instanz der DO-Klasse weder entfernt aufrufbar noch serialisierbar
    - Meist existiert zusätzliche Value-Klasse
  - Weniger Overhead als beim Entity Bean
- **Bildung von grobgranularen Komponenten**
  - z.B. Partner (Entity Bean) mit Lieferadressen (DO), Rechnungsadresse (DO), ...

9

Die EJB Query Language

- **SQL ähnliche Abfragesprache**
- **Entity Beans und DOs definieren abstraktes Persistenz-Schema**
  - Unterstützung von Navigation, Methoden-Parametern, Aufruf von Finder-Methoden,...
- **Spezifikation von Finder-/Select-Methoden im DD**
- **Beispiel**
  - Suche alle Partner, die eine Adresse in Stuttgart haben
  - `partnerHome.findByOrt(„Stuttgart“);`
  - `FROM Partner p, a IN p.adressen WHERE a.ort= ?1`

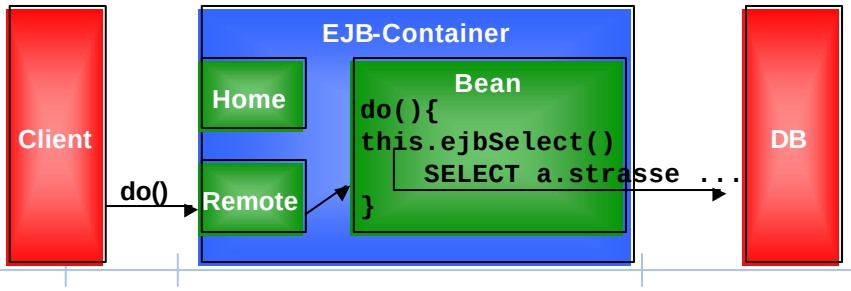


10

plenum
the architect of new business

Die EJB Query Language (2)

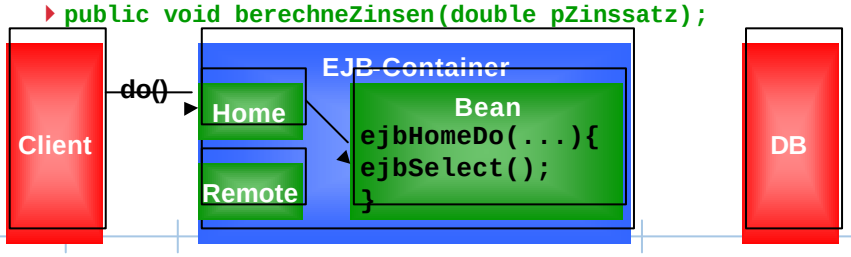
- **Select-Methoden**
  - Interne Finder-Methoden der persistenten Objekte
  - Formulierung beliebiger Suchen
- **Beispiel: Suche Strasse der Lieferadresse des Partners in Stadt**
  - `this.ejbSelectShippingAddress(„CIA“, „Langley“);`
  - `SELECT a.strasse FROM Partner p, a IN p.adressen WHERE a.ort=?2 AND p.name=?1`
- **ejbFind und ejbSelect implementiert der PMP**



plenum
the architect of new business

Die Erweiterung des Home-Interfaces

- **Bisher: nur create() und findXXX() Methoden zulässig**
- **Definition beliebiger, sogenannter Home-Methoden möglich**
  - Implementiert als ejbHome<Methode> durch Bean-Klasse
  - Ausführung auf Bean-Instanz ohne Identität (im Pooled-Zustand, wie finder-Methoden)
  - Nur für Entity Beans
- **Beispiel:**
  - Zinsgutschriften auf allen Festgeldkonten durchführen
  - `public void berechneZinsen(double pZinssatz);`



plenum
the architect of new business

Weitere Änderungen im Überblick

- **Deployment Descriptor angepaßt**
- **Unterstützung von RMI/IIOP verpflichtend**
- **EJB 1.1 CMP muß weiterhin unterstützt werden**
- **Transaktionsmanager über „java:pm/Transactionmanager“ bei JNDI registriert**
  - Notwendig z.B. für den PMP um Persistenzframework zu integrieren
- **Stärkung der Interoperabilität zwischen Application Servern**
  - RMI/IIOP
  - ...
- **Bugfixing**
- **Klarstellungen**

13

plenum
the architect of new business

Verfügbare Implementierung von EJB 2.0

- **J2EE Referenzimplementierung 1.3 beta 1**
  - Vollständige EJB 2.0 Implementierung
  - Problem: Deploytool extrem fehlerträchtig
- **Bea Weblogic Server 6.0**
  - Basiert auf Frühversion von EJB 2.0 (ohne DO)
  - Problem: webbasiertes Deploytool recht langsam (Pentium III 400 256 MB) und fehlerträchtig
- **Vielfach nur teilweise Implementierungen**
  - Meist mit MessageDriven Beans ohne EJB 2.0 CMP

14

plenum
the architect of new business

Bewertung von EJB 2.0

- **Fortschritte**
  - Benutzbarkeit von Entity Beans
    - Grobranularere Komponenten
    - Automatische Beziehungs-verwaltung
  - Theoretisch jedes Persistenzframework benutzbar
  - Einheitliche Abfragesprache
  - Asynchrone Kommunikation
- **Todos**
  - Vererbung von Komponenten (insb. Entity Beans)
  - Definition von Standards für Deployment Werkzeuge
  - Definition wie Persistence Manager in AS und dessen Werkzeuge integriert werden
- **Gesamturteil**
  - Aus dem Baby EJB ist ein Teenager geworden.

15

plenum
the architect of new business

Vielen Dank für Ihre Aufmerksamkeit

16