

## Rollenmuster



### Zweck

“Modelliere die verschiedenen Blickwinkel eines fachlichen Gegenstands in eigenen Objekten, den sog. *Rollenobjekten*. Diese Rollenobjekte können dynamisch zu *Kernobjekten* hinzugefügt und von diesen entfernt werden. Das Kernobjekt zeigt die Merkmale, die ein fachlicher Gegenstand in jedem Kontext oder Bereich ausweist. Stelle die gemeinsame fachliche Identität von Kern- und Rollenobjekten sicher.”

### Anwendbarkeit

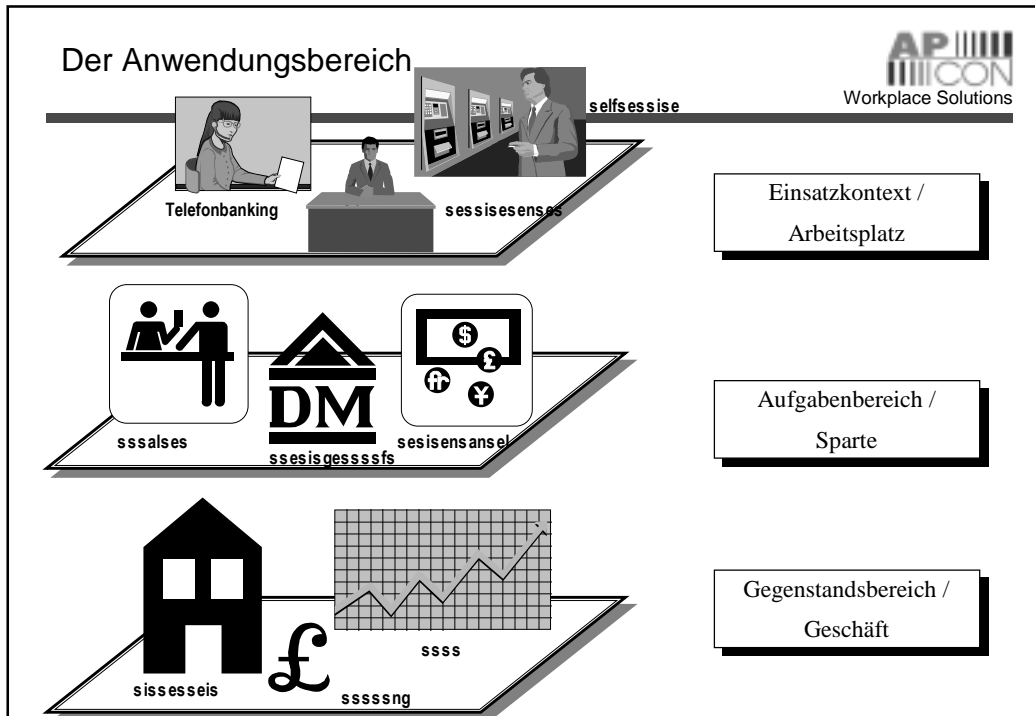
- Kernkonzepte sollen in unterschiedlichen Kontexten mit spezifischen Schnittstellen und Attributen versehen werden.
- Die kontextspezifischen Blickwinkel müssen dynamisch zu einem Kernobjekt hinzugefügt und auch wieder von diesem entfernen werden.
- Das aus Kern- und Rollenobjekt bestehende Objektgefüge soll über eine eigene Subjektidentität verfügen.
- Die verschiedenen Blickwinkel (Rollen), unter denen ein Kernkonzept betrachtet werden kann, müssen unabhängig voneinander weiterentwickelbar sein.
- Das Muster sollte **nicht** verwendet werden, wenn zwischen den Rollen starke Abhängigkeiten bestehen.

## Kontext: Ein objektorientierter Schichtenarchitekturstil



Workplace Solutions

- Auf der Grundlage der Struktur des Anwendungsbereichs lässt sich folgender objekt-orientierter Schichtenarchitekturstil definieren:
  - *Einsatzkontext* und *Aufgabenbereich*: entsprechenden den Kontexten, die bereits für das fachliche Modell definiert worden sind.
  - *Gegenstandsbereich*: Er stellt die fachliche Basis für die auf ihr konstruierten Frameworks und Anwendungssysteme bereit. Die Komponenten dieser Schicht ermöglichen die fachliche Integration der in den Aufgabenbereichen entwickelten Frameworks.
  - *Technologiebereich*: Komponenten dieser Schicht vergegenständlichen die bei der Softwareentwicklung eingesetzten Modelle über die verwendete Technik. Er stellt somit die technische Kombinierbarkeit der entwickelten fachlichen Komponenten sicher.
  - *Systembasis*: Kapselt die durch die Systemarchitektur und der Entwicklungsumgebung zur Verfügung gestellten Schnittstellen.



## Einteilung des Anwendungsbereichs

### Einsatzkontext / Arbeitsplatz

Aus Kundensicht werden die Dienstleistungen eines Unternehmens an konkreten Arbeitsplätzen angeboten. In kundenorientierten Unternehmen wie Banken sind dies zunehmend sog. Servicecenters. Klassische Arbeitsplätze sind dagegen z.B. Schalter oder Devisenhandel. Diese konkreten Organisationsformen der Arbeit nennen wir (je nach Sicht) Einsatzkontext oder Arbeitsplatz.

### Aufgabenbereich / Sparte

Die Dienstleistungen eines Unternehmens werden fachlich in Sparten eingeteilt: Kreditgeschäft, Anlagengeschäft, Kontoverwaltung etc. Diese Sparten nennen wir Aufgabenbereiche. Traditionell ist ein Unternehmen entlang dieser Sparten in Abteilungen aufgeteilt. Heute gibt es zumindest Spezialisten für jeden Aufgabenbereich.

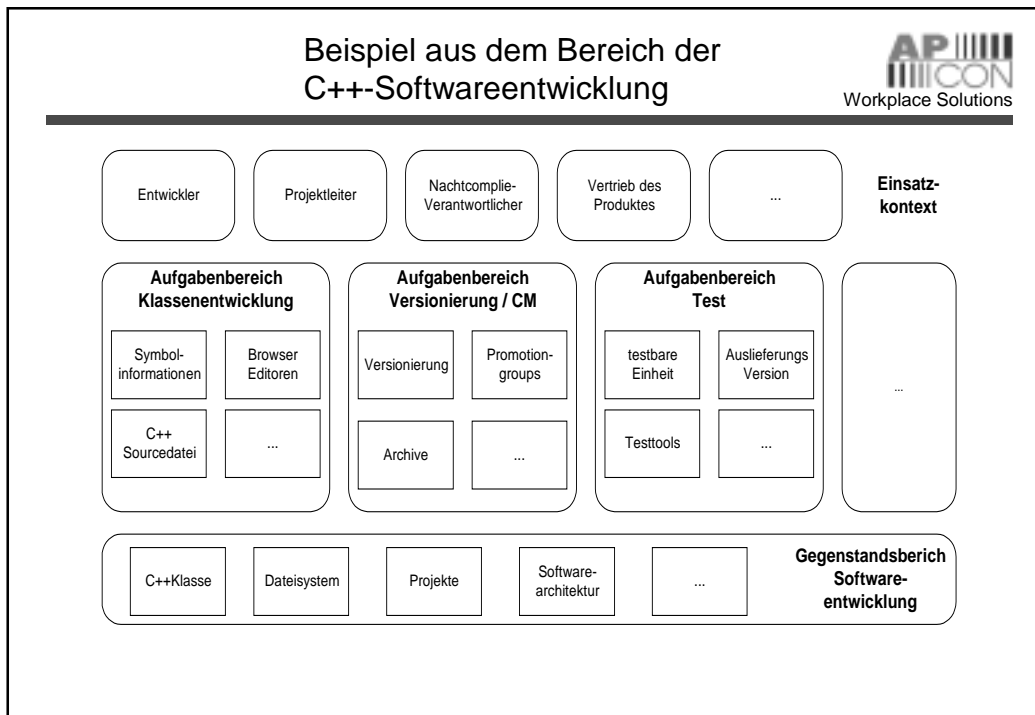
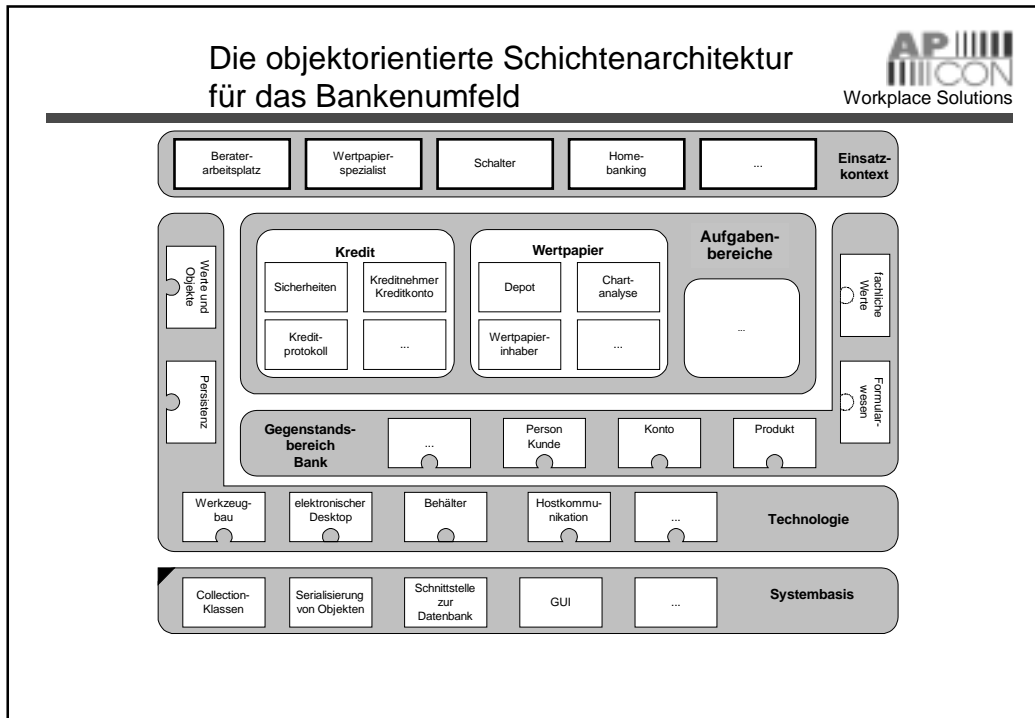
### Gegenstandsbereich / Geschäft

Zwischen den Aufgabenbereichen gibt es Überlappungen, d.h. Konzepte und Begriffe, die in (fast) allen Aufgabenbereichen verwendet werden (z.B. Kunde, Konto oder Zinssatz). Diese übergreifenden Begriffe und Konzepte ermöglichen die Identifikation eines gemeinsamen Gegenstands unabhängig vom jeweiligen Anwendungsbereich. Wir bezeichnen den Bereich der überlappenden Begriffe und Konzepte als Gegenstandsbereich. Der Gegenstandsbereich beschreibt das "Geschäft" eines Unternehmens

# Das Rollenmuster

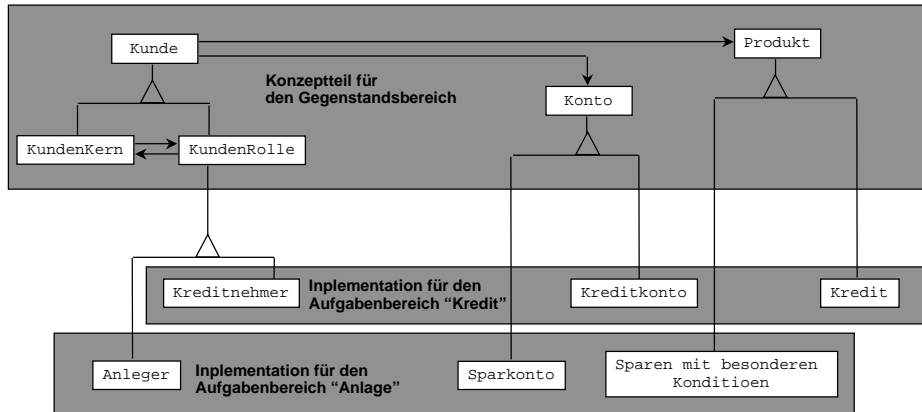


Workplace Solutions



## Beziehungen zwischen den Layers - am Beispiel -

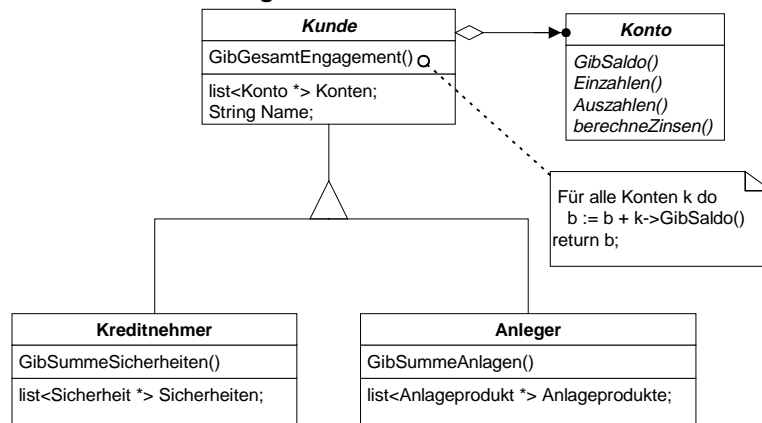
- Die Rahmenwerken der **Aufgabengebiete** bauen auf den Rahmenwerken des **Gegenstandsbereichs** auf.
- Der Layer für den **Gegenstandsbereich** sollte stabil gehalten werden, da alle anderen Layer auf ihm aufbauen.



7

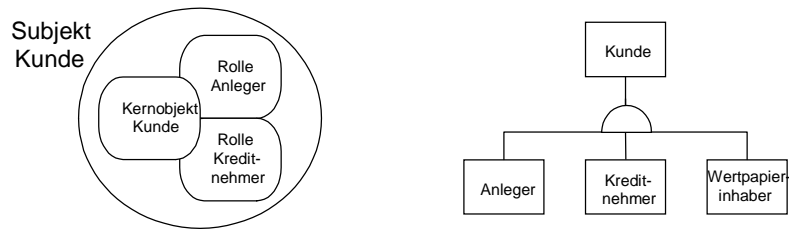
## Produktbereichsspezifische Spezialisierung eines Kunden mittels Vererbung

- die naive Lösung -

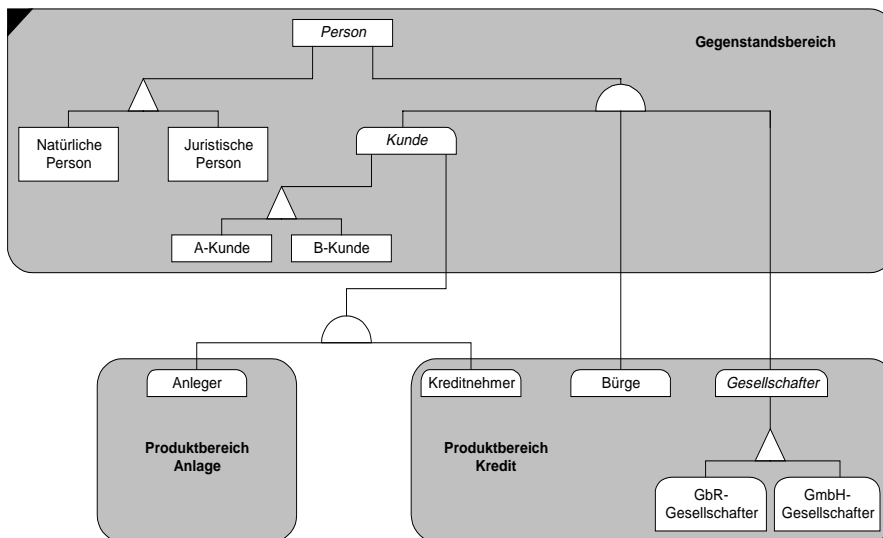


## Rollen als Modellierungsmittel

•Die Modellierung eines Anwendungssystems wird immer von einem ganz bestimmten Blickwinkel aus durchgeführt (z.B. aus der Sicht einer Kreditverwaltung). Rollen ermöglichen es, die aus der Sicht eines Aufgabenbereichs fachlich zusammengehörigen Attribute und Umgangsformen eines Konzepts aus dem Gegenstandsbereich in einer eigenen Klasse zu modellieren.



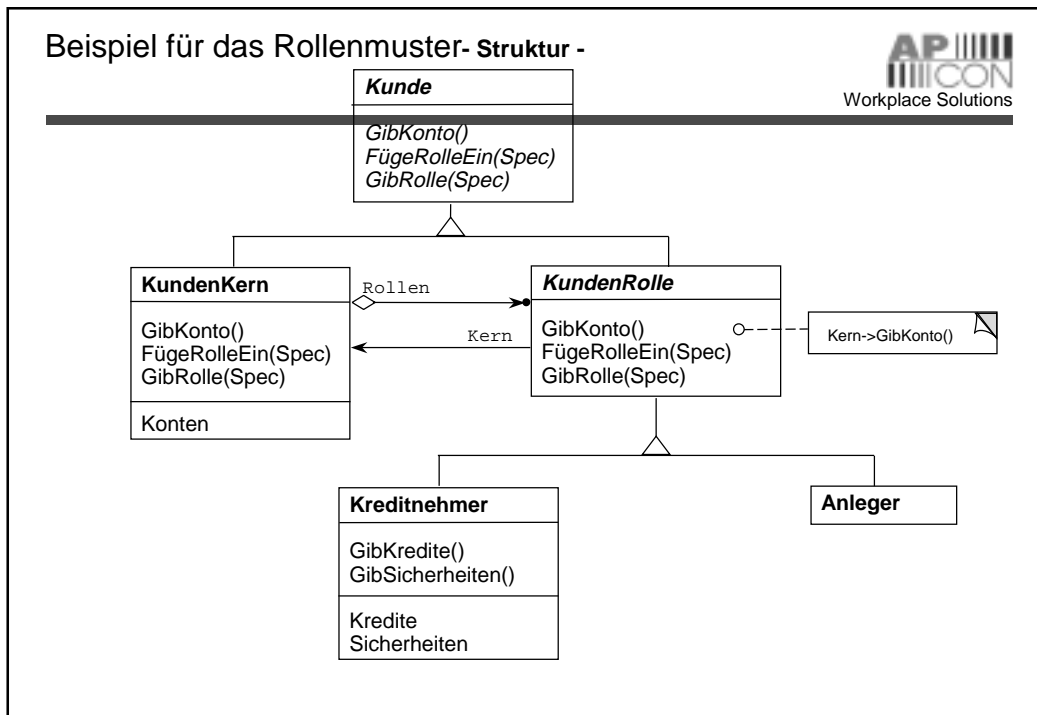
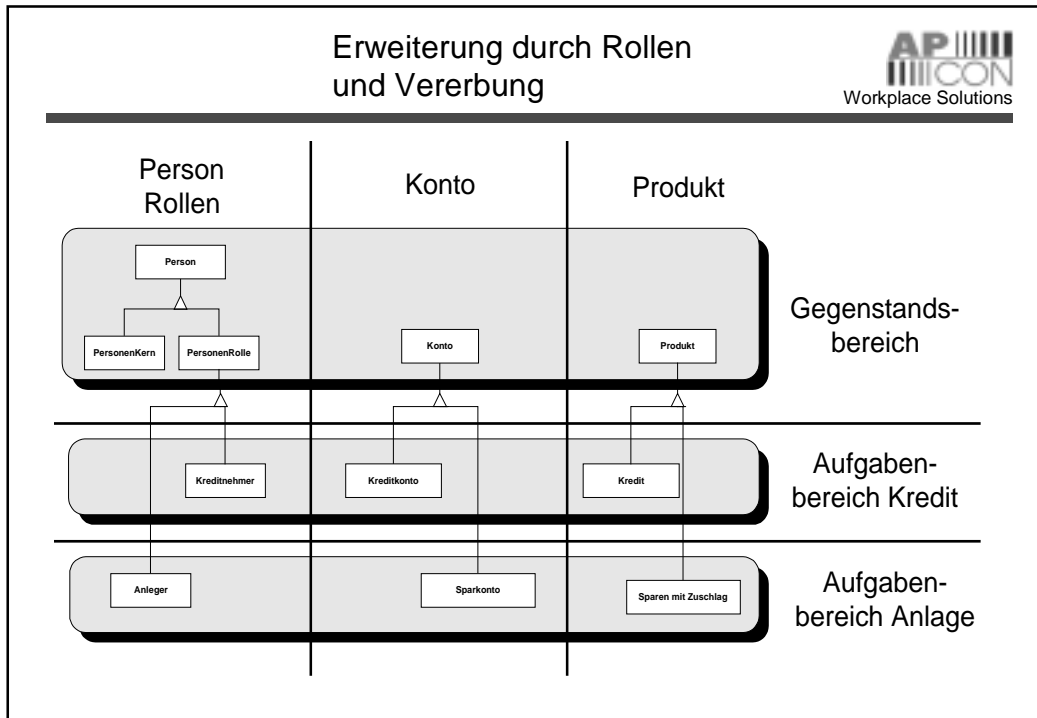
## Eine Rollenhierarchie



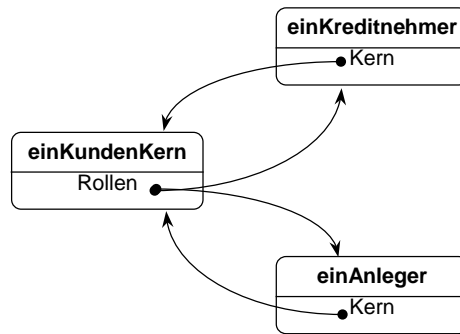
# Das Rollenmuster



Workplace Solutions

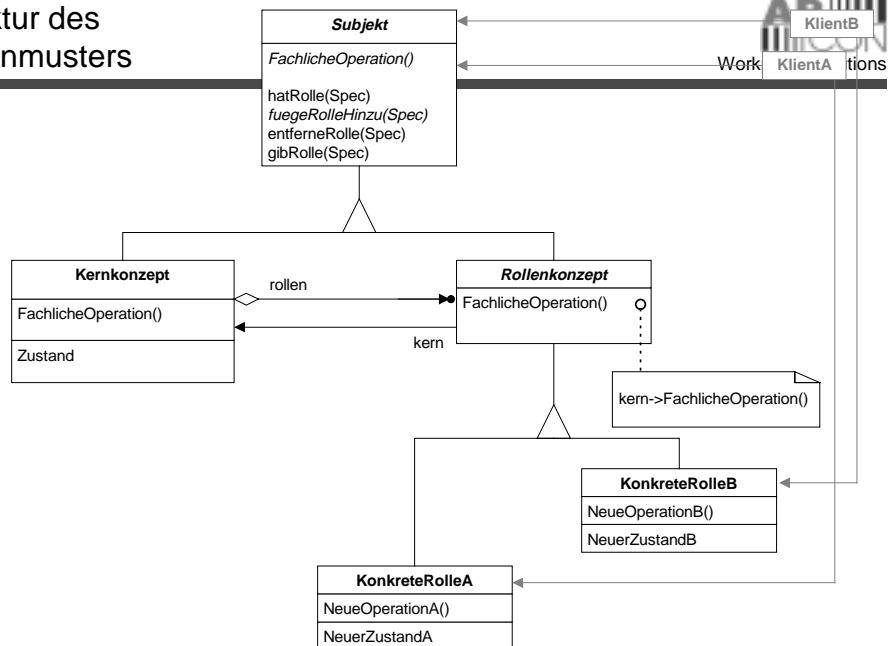


## Beispiel für das Rollenmuster - Dynamik -



13

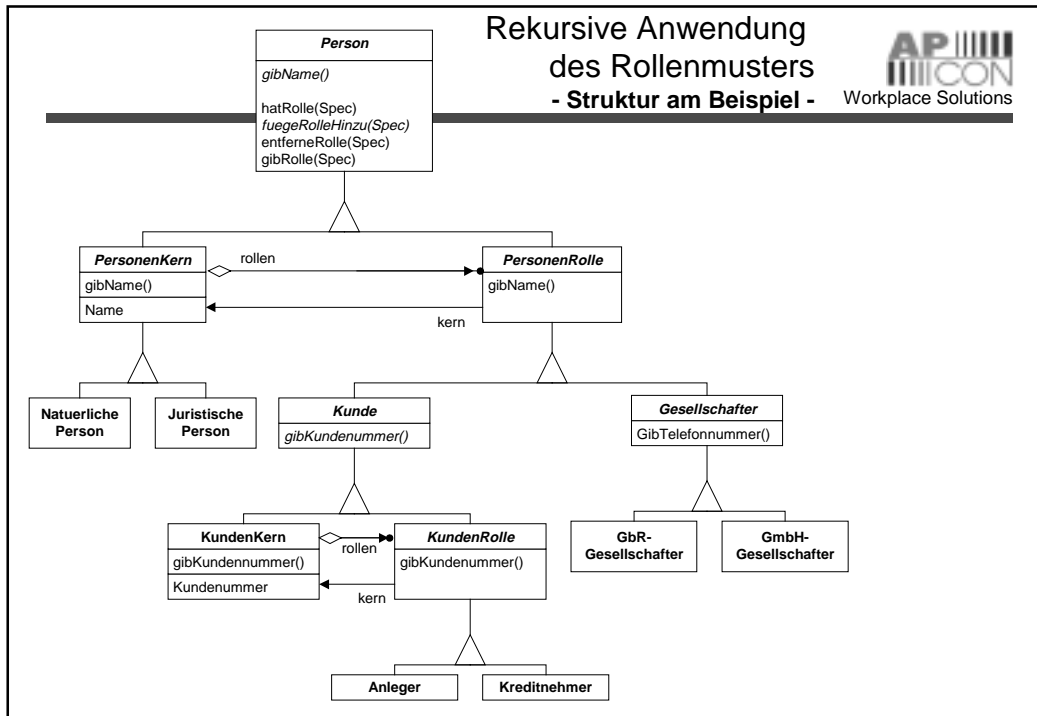
## Struktur des Rollenmusters



# Das Rollenmuster



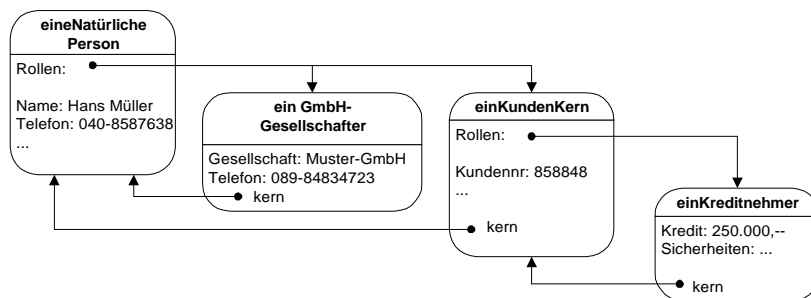
Workplace Solutions



### Rekursive Anwendung des Rollenmusters - Dynamik am Beispiel -



Workplace Solutions





## Diskussion des Rollenmusters

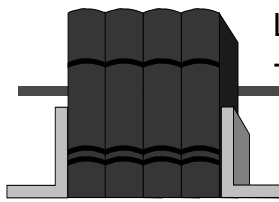
### - Vorteile -

- Das Kernkonzept verfügt über eine **hohe Kohäsion**, da es keine Schnittstellen und Attribute enthält, die nur aus einem bestimmten Blickwinkel (z.B. Produktbereich) motiviert sind.
- Rollenkonzepte können **unabhängig** voneinander entwickelt werden, da hierfür keine Änderungen am Kernkonzept notwendig sind.
- Rollenobjekte können **dynamisch** zu Kernobjekten hinzugefügt und von diesen entfernt werden. Zur Laufzeit eines Anwendungssystems befinden sich nur die Objekte im **Hauptspeicher**, die auch tatsächlich verwendet werden.
- Im Rahmen unserer **Modellarchitektur** können Produktbereiche in ihrer Entwicklung besser **voneinander getrennt** werden, da Veränderungen, die aus Sicht eines Produktbereichs an einer Rolle vorgenommen werden, keine Auswirkungen auf Rollenkonzepte anderer Produktbereiche haben.

## Diskussion des Rollenmusters

### - Nachteile -

- Regeln, die sich auf das Zusammenspiel von Rollenobjekten mit ihrem Kernobjekt beziehen, können **nicht durch das Typsystem überprüft** werden. Ist beispielsweise das Hinzufügen einer Rolle von der Existenz einer anderen Rolle abhängig, so kann dies nur zur Laufzeit des Systems abgesichert werden.
- Das Rollenmusters erfordert **zusätzlichen Kodierungsaufwand** beim Klienten. Klienten müssen i.d.R. zunächst von einem Kernobjekt das **passende Rollenobjekt** anfordern. Ist dies noch nicht vorhanden, so muß es vom Klienten erzeugt werden. Dabei muß der Klient fachlich sicherstellen, daß das Kernobjekt diese Rolle tatsächlich spielen kann. Erst dann läßt sich das Subjekt vom Klienten in der gewünschten Rolle ansprechen.
- **Zusätzlicher Kodierungsaufwand** ist auch notwendig, um die **Subjektidentität** zu realisieren. Das Subjekt, das Kern - und das Rollenkonzept müssen daher über entsprechende Methoden verfügen, um die Subjektidentität auf die technische Objektidentität von Kern- und Rollenobjekt abzubilden.



## Literaturhinweise

### - Strukturierung von Rahmenwerken -



Workplace Solutions

- E. Gamma:** *Objektorientierte Softwareentwicklung am Beispiel ET++*. Springer, 1992.
- W. Bischofberger et al.:** *Wiederverwendung durch Framework-Technik - vom Mythos zur Realität*. OBJEKTSpektrum 5, 1995.
- S. Cotter, M. Potel:** *Inside Taligent Technology*. Addison-Wesley, 1995.
- Wolfgang Pree:** *Design Patterns for Object-Oriented Software Development*, Addison-Wesley, 1994.
- D. Bäumer:** *Softwarearchitekturen für die rahmenwerkbasierete Konstruktion großer Anwendungssysteme*. Dissertationsschrift am Fachbereich Informatik der Universität Hamburg, Januar 1998.
- D. Bäumer, G. Gryczan, R. Knoll, C. Lilienthal, D. Riehle, H. Züllighoven:** *Framework Development for Large Systems*. Communications of the ACM, October 97, Vol. 40, No. 10.
- H. Züllighoven:** *Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material-Ansatz*. Heidelberg: dpunkt-Verlag, 1998.