

# **Vorlesung Software aus Komponenten**

## **1. Komponenten – Markt – Standards**

Prof. Dr. Hans-Gert Gräbe  
Wintersemester 2011/12

### Komponentenmarkt

Ansatz brummt richtig, wenn die erforderlichen Komponenten in guter Qualität und ausreichend breiter Funktionalität verfügbar sind

- Modell Komponentenmarkt vs. Open Source und GPL

Komponentenmärkte existieren (erst) in Ansätzen

- Warum Mehrzahl?
- technologische und ökonomische Rahmenbedingungen erforderlich
  - technologische Bedingungen existieren seit Ende der 60er Jahre
  - Markt erfordert Standard, da Komponenten nur in einer entsprechenden Infrastruktur (Komponentenplattform) operieren können. Standardsetzung ist ein politisches Problem
- Beispiel: Markt für VBX

Warum ist Komponenten-Software auch in Prä-Marktpphase attraktiv?

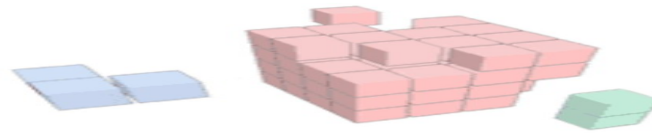
- Vorbereitung auf kommenden Markt
- Übergang zu komponentenbasierter Software ist aufwändig
- Setzen von Standards und Sammeln von Kompetenz

Die selbstverstärkende Wirkung eines Komponenten-Markts

Wird es wirklich dauerhaft Komponenten-Märkte geben?

- Bindung an Standards einer gewissen IT-Generation
- Infrastruktureller Charakter einer Software-Komponenten-Szene
- Ökonomische Besonderheiten von Software und deren inhärente Tendenz zur Monopolisierung

## 4 Haupteigenschaften von Komponenten



... eine funktional und  
technisch abgeschlossene  
ausführbare Einheit

... unabhängig als Einheit  
entwickel- und  
konfigurierbar

Eine Komponente ist...

... wiederverwendbar

... nur durch genau  
spezifizierte Schnittstellen  
ansprechbar

## Baustein-Charakter

- „Alle Welt ist aus Bausteinen, nur die Software-Branche nutzt dieses Konzept noch nicht“
- Problem: Bauplan- und Metaprodukt-Charakter von Software
  - Unterscheidung zwischen Software und deren Instanzen
  - Unterscheidung zwischen Blaupausen und Produkten

## Komponenten als Einheiten der Packung (deployment)

- beyond object oriented programming [Szyperski 02]
- OO has failed but component software is succeeding [Udell 94]

## Erfolgsgeschichten von Komponenten-Software

- die älteste: moderne Betriebssysteme
- Datenbanken und Transaktionsmonitore
- Plugin-Architektur (nicht nur) moderner Browser
- moderne Applikationsserver

## Gemeinsamkeiten der Erfolgsgeschichten

### Existenz einer **Infrastruktur**

- grundlegende Funktionalität für Interoperabilität wird in ausreichendem Umfang zur Verfügung gestellt

Komponenten haben **genügend substanzielle Funktionalität**, die eine wiederholte Entwicklung uneffektiv macht

Komponenten unabhängiger Anbieter können in der Infrastruktur koexistieren

- Zusammensetzbarkeit ist eher wahrscheinlich als garantiert
- plug and play

Komponenten existieren auf einer Abstraktionsebene, die eine **direkte Bedeutung** für den verteilenden Client haben

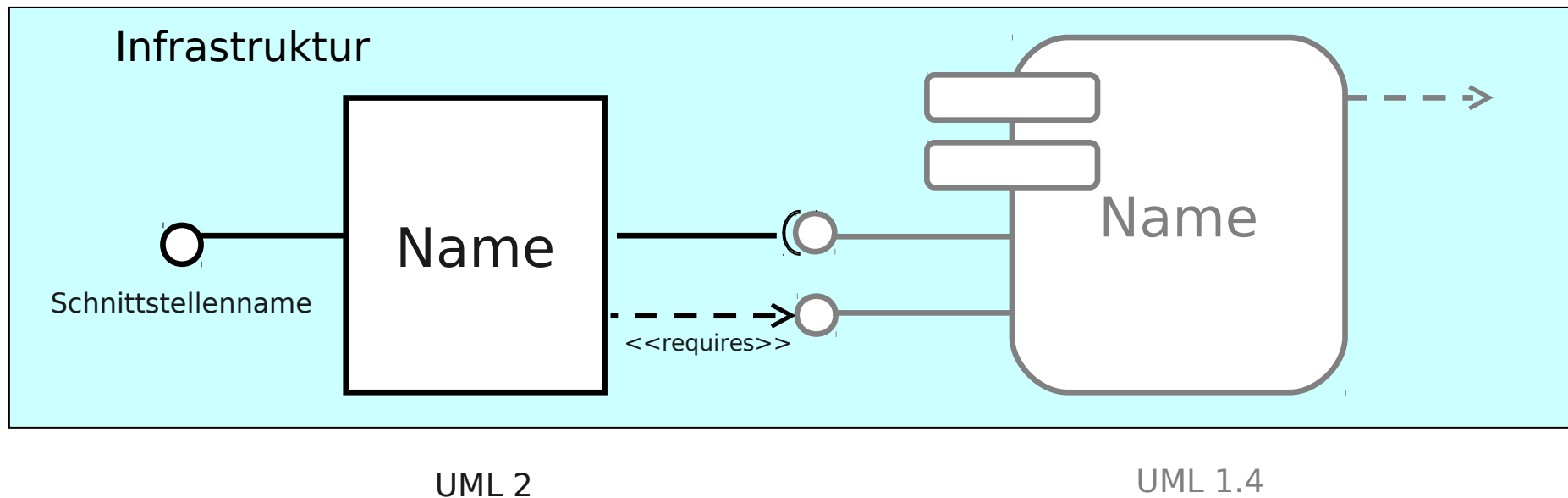
- Bsp. VB Control hat direkte grafische Repräsentation
- im Gegensatz zu Objekten, die für Nichtprogrammierer keine eigenständige Bedeutung haben
- aber: Objekttechnologie als der beste Weg zur Realisierung von Komponententechnologie

# 1.3. Komponenten

## Aufbau einer Komponente

### Schnittstellen

- Operationen / Funktionalität (funktionale Kopplung)
- Attribute / Signatur (Datenkopplung)
- Veröffentlichte Schnittstellen (exportierte Dienste)
- Konsumierte Schnittstellen (importierte Dienste)



# 1.3. Komponenten

## Komponentenrelevante Konzepte

### Komponentenrelevante Konzepte

#### Komponenten

- **Gekapselte**, generalisierte **Softwareobjekte**, die einen **Dienst** zur Verfügung stellen und aus denen größere Komponenten oder Systeme gebaut werden können

#### Kapselung

- wohlspezifizierte Dienste-Schnittstelle, Kontextunabhängigkeit

#### Generalisierung

- Parametrisierung, Erweiterbarkeit, Nutzbarkeit in unterschiedlichen Anwendungen

#### Dienst

- Zusammenhängende Sammlung von in Beziehung stehender Funktionalität. Nachfrage nach Dienst (service) von Klienten (clients)

#### Systemfähigkeit

- Kaskadierbarkeit, Katalogisierbarkeit, Aufbau nach vorgegebenen Architekturprinzipien



### Komponenten-Lebenszyklus

unterscheide zwischen

- Komponente als Konzept
- Komponente als auslieferbare prototypische Einheit
- Komponente als in einem prototypischen Systemkontext konfigurierbare Einheit
- Komponente als in einem konkreten Systemkontext zu verteilende, zu konfigurierende und zu installierende Einheit
  - unterscheide „deployed“ und „installed“
- Komponenten-Instanz als Instanz einer installierten Komponente

Komponenten und Dienste: Der Begriff „Dienst“ wird oft auch im Sinne einer Verbindung mit einem auf einem Markt positionierten Dienstanbieter verwendet

- Dienste in diesem Sinne sind orthogonal zum Komponenten-konzept und können Komponenten-Instanzen verwenden
- Dazu gehört aber eine konkrete Hardware-, Software- und Organisations-Infrastruktur

## 1.3. Komponenten Infrastruktur für Komponenten

### Infrastruktur für Komponenten

#### Komponenten-Plattform

- Hilfsmittel und Technologien, die zur Erstellung und zum Betrieb bzw. zur Anpassung flexibler und erweiterbarer Anwendungen auf der Basis von Komponenten erforderlich sind

#### Komponenten-Entwicklung

- Modelle, Methoden und Werkzeuge, die zur Analyse, Entwicklung und Design von auf Komponentensoftware beruhenden betrieblichen Anwendungssystemen dienen
- Design to / from / for component

# 1.4. Komponentenentwicklung

## Designprinzipien

### Design **for** Component

- Initiale Entwicklung atomarer Komponenten zum Ziele der Bereitstellung spezifischer, gekapselter Dienst, welche später in neue Anwendungen schnell und einfach integriert werden können.

### Design **from** Component

- Inkrementelle Entwicklung von komplexeren Komponenten und Anwendungssystemen unter Nutzung vorhandener und noch zu erstellender Bausteine, sowie der Dienste der Komponenten-Plattform.

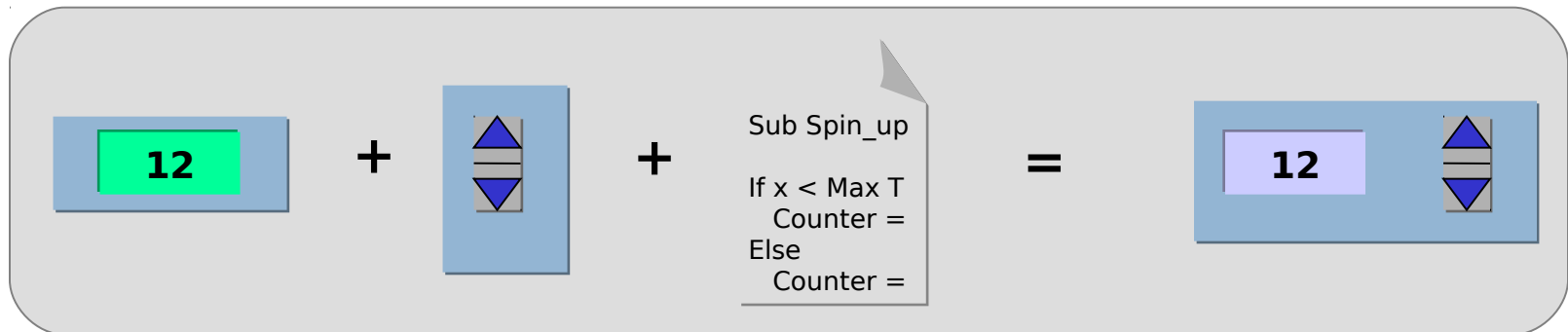
### Design **to** Component

- Methoden zur Transformation konventionell erstellter Anwendungssysteme in eine flexible komponentenbasierte Umgebung.

# 1.4. Komponentenentwicklung

## Design for Component

- Fokus: Komponente als Endprodukt
- Anbieter / Produzenten-Sicht
- Ausrichtung an:
  - Standards
  - angestrebte Zielumgebung
- Ergebnis:
  - Bereitstellung von Komponenten
  - Erfüllung einer spezifischen Aufgabe

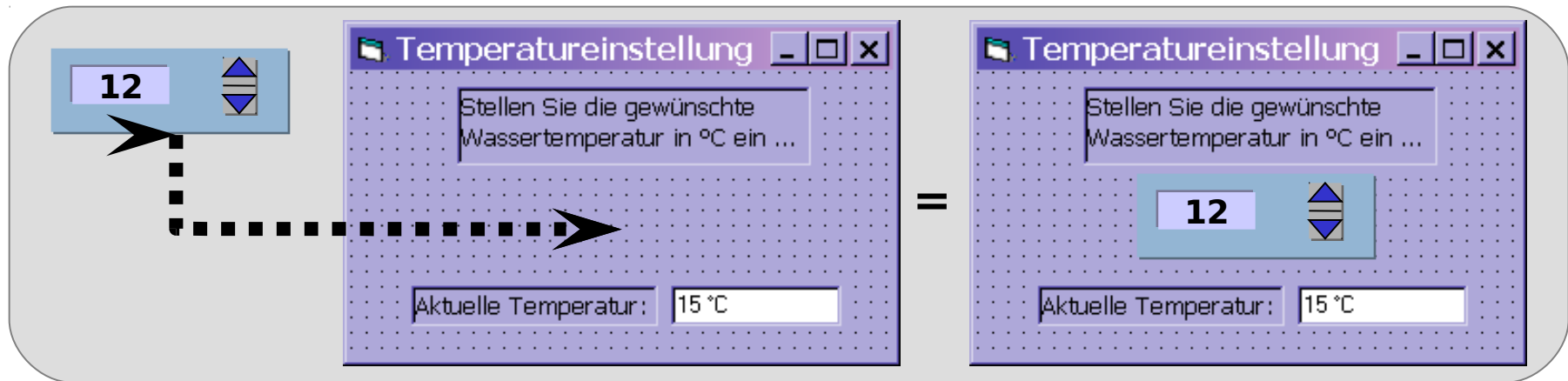


**Abb: Zusammenbau einer Komponente aus Teilkomponenten**

## 1.4. Komponentenentwicklung

### Design from Component

- Fokus: Zusammenbau komplexer Anwendungssysteme
- Käufer / Anwender-Sicht
- Aggregation von Komponenten auf höheren Niveau
- Grundlagen:
  - Spezialisierte Komponenten
  - "Komponenten-Kleber"
  - Dienste der Komponenten-Plattform
- Ergebnis:
  - Aufgabenspezifische Anwendungskomposition

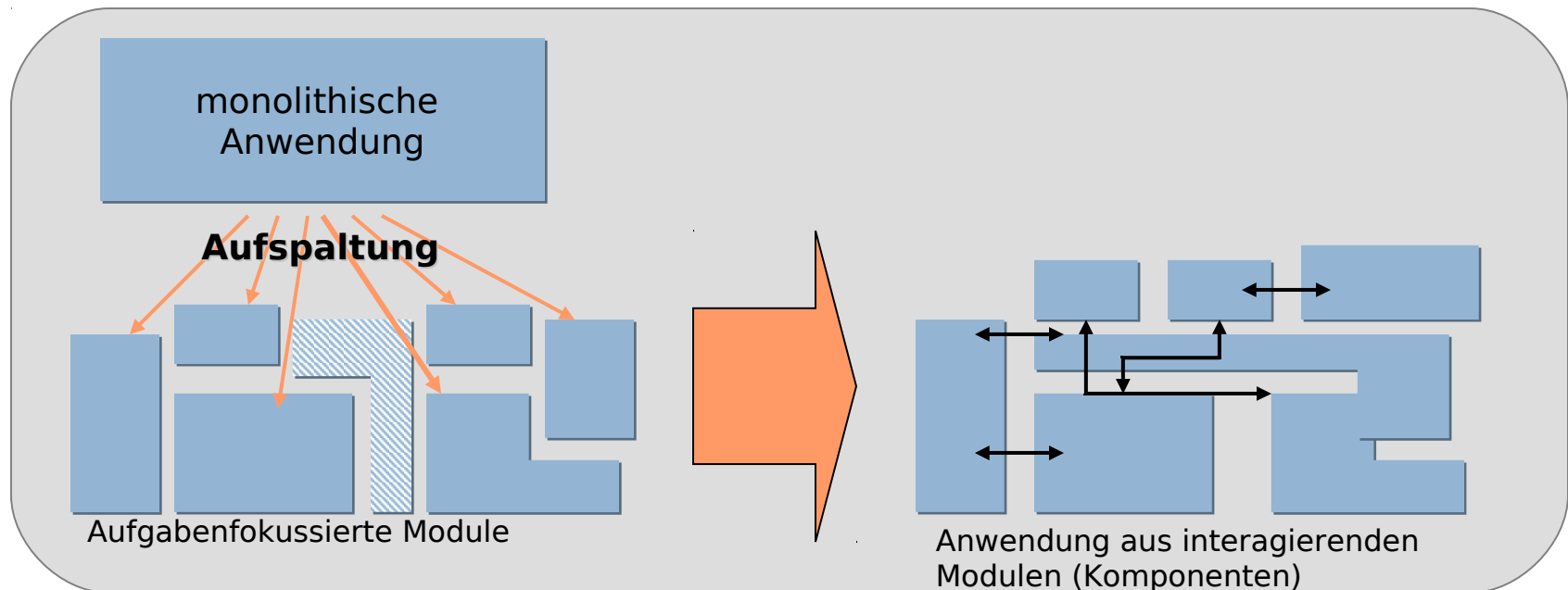


**Abb: Verwenden einer Komponente**

# 1.4. Komponentenentwicklung

## Design to Component

- Migration zu komponentenbasierten Lösungen
- Anwendungs-Reengineering
  - "Zerschlagen" von Altanwendungen
  - Monolithische Altanwendung wird zu einer flexiblen neuen komponentenbasierten Anwendung umgebaut
- Verwendung vorhandener Komponenten
- Ersetzung alter Komponenten durch neue Komponenten



**Abb: von einer monolithischen zu einer modularen Anwendung**