



Softwareentwicklung mit Komponenten

Fachhochschule Gießen

20.10.1999

Dr. Burkhardt Renz



Gliederung

– was sind unsere Ziele?

- ▶ Was ist eine Komponente?
Beispiele, Charakteristika, Technologien
- ▶ Wie funktioniert das?
Ein konkretes Beispiel mit COM/ActiveX
- ▶ Wiederverwendung und Komponenten
Softwaretechnik der Zukunft?



Komponententechnologie **– wir alle verwenden sie**

- ▶ **Fahrscheinautomat von ASCOM**
Software-Farm für wiederverwendbare Komponenten
- ▶ **Macintosh Extensions, Netscape Plug-ins**
Funktionalität erweitern oder verändern
- ▶ **Sybase/Powersoft's DataWindow**
Herzstück in PowerBuilder, Java und C++
- ▶ **Verbunddokumente (Compound Documents)**
Exceltabelle im Word-Dokument
- ▶ **Elemente graphischer Oberflächen**
die Erfolgsstory von Visual Basic und JavaBeans

1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Komponenten

– wodurch sind sie charakterisiert?

- ▶ Eigenständige Einheit eines Softwaresystems
hohe Kohäsion, geringe Kopplung
- ▶ Kann und soll zusammengesetzt werden
unterstützt Infrastruktur zur Komposition
- ▶ Klar spezifizierte Schnittstelle
Kontrakt zwischen Verwender und Komponente
- ▶ Explizite Kontextabhängigkeit
Umfeld der Komponente definiert
- ▶ Verwendung unabhängig von Entwicklung
Black Box, Sprach- und Plattformunabhängig
- ▶ Implementierung änderbar, Schnittstelle fix
Kapselung ohne den Verwender zu kennen

1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Interfaces

– was gehört zum Kontrakt?

▶ Signatur der Funktion

```
interface IWebBrowser : IDispatch
{
    HRESULT Navigate( [in] char* URL )
}
```

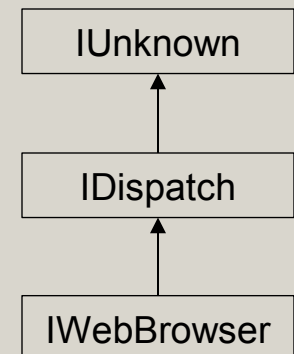
▶ Vor- und Nachbedingungen

```
pre URL != NULL
post URL ist invariant
```

▶ Ersetzbarkeit der Komponente

Polymorphismus, Typisierung von Interfaces

▶ Ableitung von Interfaces



1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Technologien

– wie verbindet man Komponenten?

- ▶ CORBA Common Object Request Broker Architecture

Standardisierung von Protokoll/Architektur



- ▶ COM/DCOM Distributed Component Object Model
Binärstandard



- ▶ JavaBeans, Enterprise JavaBeans
Standardisierte Java-Interfaces



1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Alltag der Softwarewerkstatt I – die Aufgabenstellung

- ▶ Dokumentenaustausch zwischen Behörden
Standesamt meldet Geburt an Einwohnermeldeamt
- ▶ Softwaretechnisches Umfeld
Dokument in XML, Programm mit MFC
- ▶ Unsere Aufgabe
Anzeige des XML-Dokuments
- ▶ Randbedingung
sollte gestern schon fertig sein
- ▶ Konzepte? Lösungen?

1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Alltag der Softwarewerkstatt II – Grundzüge der Implementierung

- ▶ Die Komponente: Microsoft WebBrowser
Internet Explorer 5 kann XML

- ▶ Verwendung der Komponente I
Ableitung von der Fensterklasse CWnd

```
class CWebBrowser : public CWnd { ... };
```

- ▶ Verwendung der Komponente II
WebBrowser erzeugen und auf URL navigieren

```
Create( CLSID_WebBrowser );  
Navigate( "giessen.intranet.standesamt" );
```

- ▶ Verwendung der Komponente III
Reaktion auf Ereignisse einbauen

```
OnDownloadBegin() { ... }  
OnDownloadComplete() { ... }
```

1 Was ist eine Komponente?

2 Wie funktioniert das?

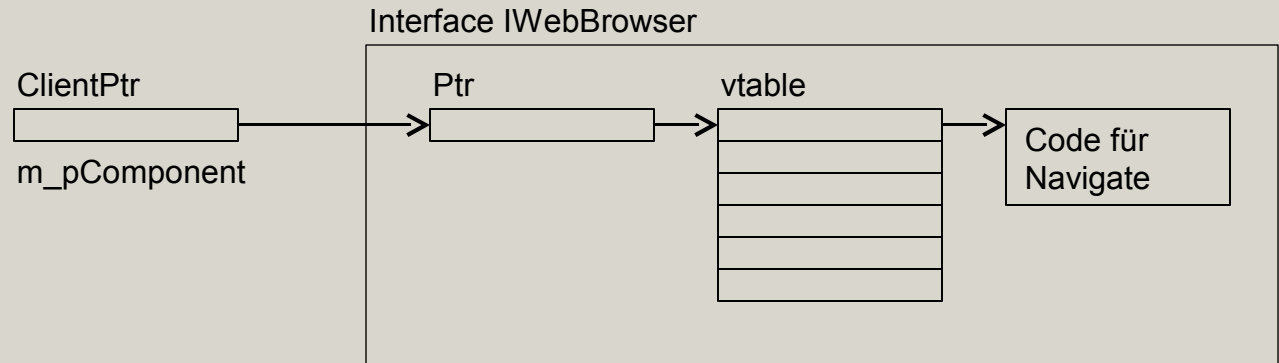
3 Wiederverwendung und Komponenten



Komponenten am Beispiel COM – wie wird eine Methode aufgerufen?

▶ Prinzip

ein Pointer auf eine Tabelle von Funktionspointern



▶ Transparenz

In-Process Server (DLL), Local Server (EXE) oder Remote Server durch rpc

1 Was ist eine Komponente?

2 Wie funktioniert das?

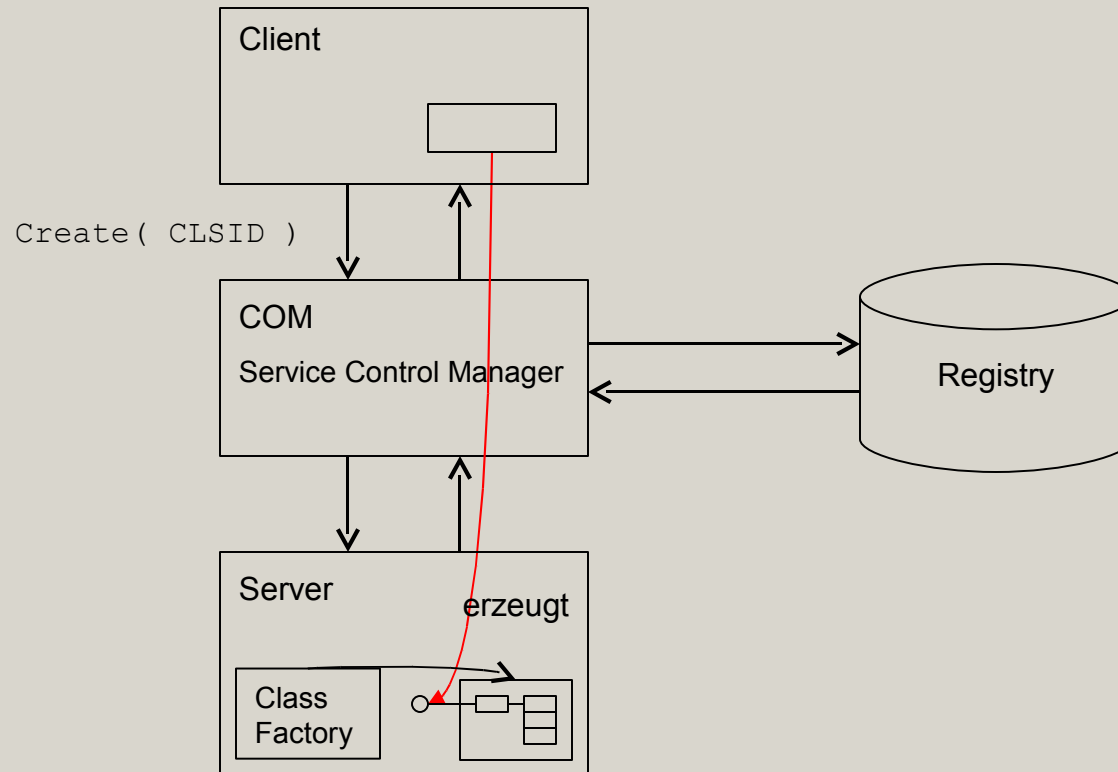
3 Wiederverwendung und Komponenten



Komponenten am Beispiel COM – wie kommt eine Instanz zustande?

▶ Prinzip

ClassID, Service Control Manager und Class Factory



- 1 Was ist eine Komponente?
- 2 Wie funktioniert das?
- 3 Wiederverwendung und Komponenten



Formen der Wiederverwendung – wo ordnen sich Komponenten ein?

- ▶ **Konzepte**
Codestandards, Algorithmen, Entwurfsmuster, Architektur
- ▶ **Quellcode**
Cut&Paste, "Klonen", Codegeneratoren
- ▶ **Binärcode in Bibliotheken, Templates**
Funktionsbibliotheken, Klassenbibliotheken, STL
- ▶ **Frameworks**
Architektur als Klassenbibliothek
- ▶ **Komponenten**
Infrastruktur austauschbarer, komplexer Teile

1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Besonderheit von Komponenten I **– aus Sicht des Verwenders**

- ▶ **Neuer Stil des Programmierens**
Visual Programming, Konzepte adaptieren
- ▶ **Wirkungen auf die eigene Anwendung**
Standards, Muster der Interaktion, Corporate Design
- ▶ **Qualitätssicherung**
keine Lebenszyklus-QS, Black Box, impliziter Massentest
- ▶ **Kosten und zeitlicher Aufwand**
Kompetenz in der Komponente verkörpert
- ▶ **Beispiel aus der Praxis**
Ultimate Grid von Dundas – mit Sicherheitsnetz

1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Besonderheit von Komponenten II **– aus Sicht des Herstellers**

- ▶ **Allgemeinheit und Anpassbarkeit**
Komponente muss für die Wiederverwendung konzipiert sein
- ▶ **Umfeld für Wiederverwendung**
Dokumentation, Testumgebung, Tutorien
- ▶ **Qualitätssicherung**
Stabilität der Schnittstelle und des Konzepts
- ▶ **Was wiederverwendbar ist, wird spät entdeckt**
je näher am Bedürfnis des Anwenders und doch in ähnlichem Kontext wiederverwendbar, desto ausgeklügelter muss die Komponente sein

1 Was ist eine Komponente?

2 Wie funktioniert das?

3 Wiederverwendung und Komponenten



Literatur

Clemens Szyperski: Component Software, Beyond Object-Oriented Programming, Addison-Wesley 1998

Frank Griffel: Componentware, Konzepte und Techniken eines Softwareparadigmas, dpunkt.verlag 1998

David J. Kruglinski, George Shepherd, Scot Wingo: Programming Microsoft Visual C++, Microsoft Press 1998

Don Box: Essential COM, Addison-Wesley 1998

Links

CORBA <http://www.omg.org>

COM <http://www.microsoft.com/com>

JavaBeans <http://www.javasoft.com>