



Komponentenfindung in monolithischen objektorientierten Anwendungssystemen

Johannes Maria Zaha, **Stephan Kelch**



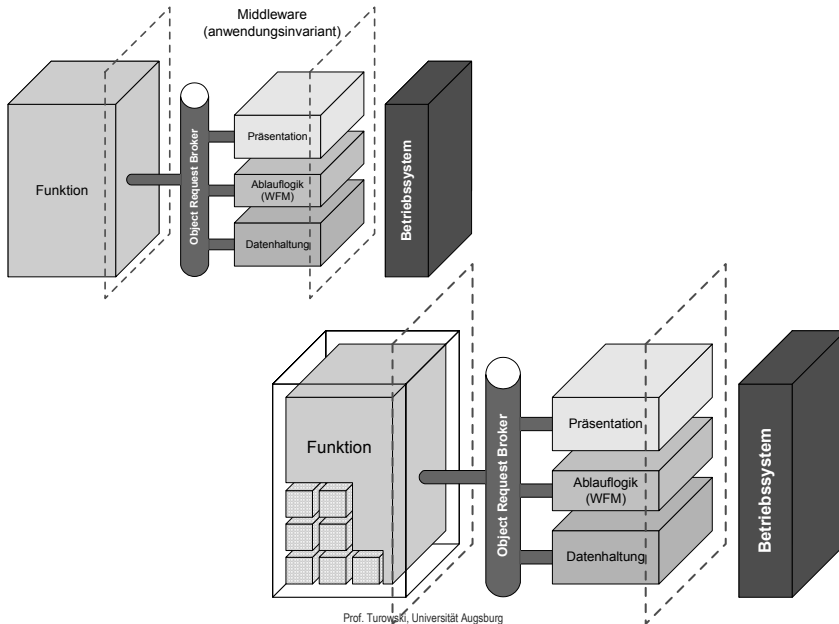
Lehrstuhl für Betriebswirtschaftslehre,
Wirtschaftsinformatik und Systems Engineering
Universität Augsburg
Universitätsstraße 16, 86159 Augsburg
Tel.: +49 (821) 598-4431, Fax -4432
URL: <http://www.wi-se.org>



Agenda

- Ausgangsproblematik
- Lösungsalgorithmus
 - ▶ Abbildung der funktionalen Struktur
 - ▶ Abbildung der Objektstruktur
 - ▶ Funktionale Zuordnung der Klassen
 - ▶ Identifikation der Frameworks / Zuordnung der Superklassen
 - ▶ Identifikation der Superklassen
- Zusammenfassung und Ausblick

Monolithen und Komponentenfindung



Ansatzpunkt

■ Zwei Strukturen innerhalb eines Anwendungssystems

- ▶ Logische Struktur
- ▶ Objektorientierte Struktur

⇒ Ziel: Zusammenführen der Strukturen

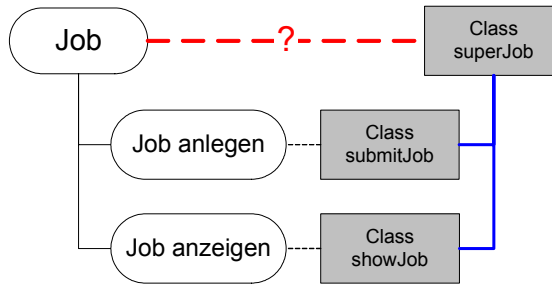
■ Lösungsansatz

- ▶ Funktionale Zerlegung des Anwendungssystems
- ▶ Darstellung der Objekthierarchien
- ▶ Gegenüberstellung der beiden Strukturen
- ▶ Elementarfunktionale Zuordnung der Klassen / Objekte
- ▶ Identifikation der Komponenten-Frameworks und der Fachkomponenten

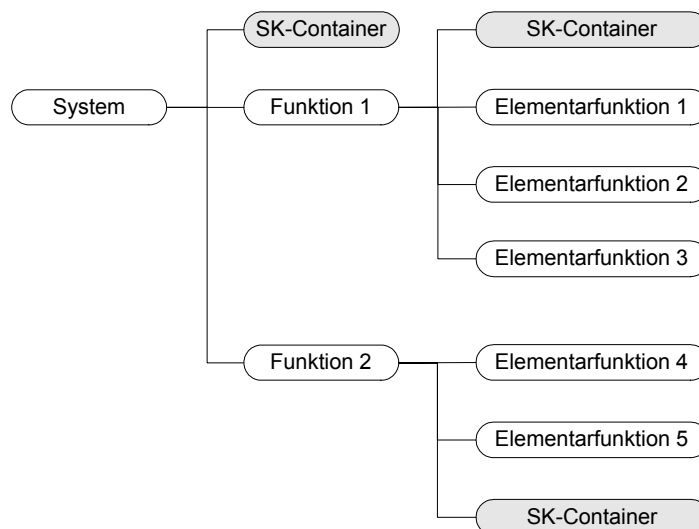
⇒ Problem: Einige Klassen lassen sich nicht elementarfunktional zuordnen

Vererbung in der Objektorientierung

- Einzelne Klassen können von der übergeordneten Superklasse Eigenschaften / Methoden erben
- Superklassen sind u. U. nicht eindeutig zuzuordnen

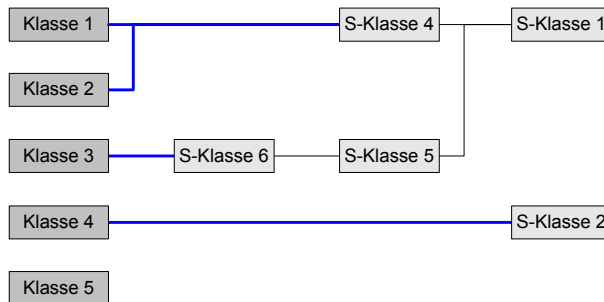


Schritt 1: Funktionale Zerlegung



Schritt 2: Abbildung der Objektstruktur

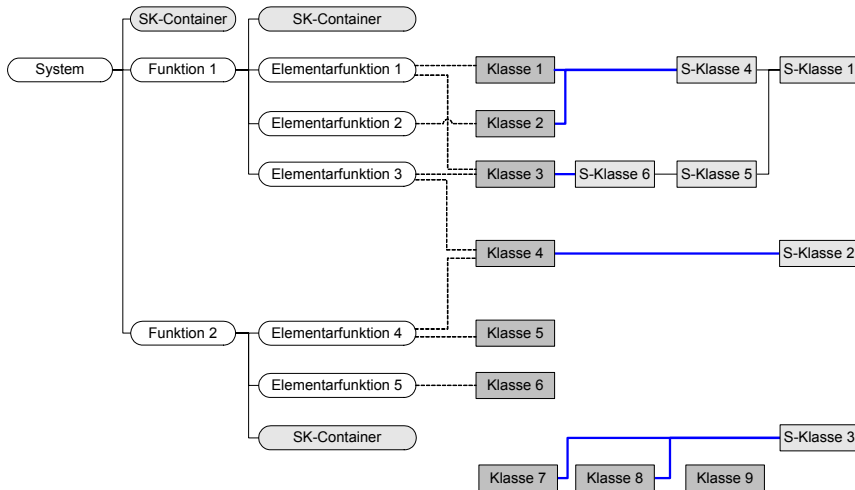
- Abbildung der Vererbungsbeziehungen der Klassen untereinander in einem Stammbaum
- Begonnen wird mit den Superklassen, sind alle Superklassen abgebildet werden die restlichen Klassen zugeordnet



Schritt 3: Funktionale Zuordnung der Klassen

- Eindeutig elementarfunktionaler Bezug
 - ▶ Eine Klasse kann direkt einer Elementarfunktion zugeordnet werden
- Kein eindeutiger elementarfunktionaler Bezug
 - ▶ Klassen, die für die Erfüllung verschiedener Elementarfunktionen genutzt werden
 - ▶ Klassen, die zwar mehreren Elementarfunktionen zugeordnet sind, diese Elementarfunktionen aber nur einer Funktion zugeordnet sind
 - ▶ Klassen, die mehreren Elementarfunktionen zugeordnet sind, die wiederum mehreren Funktionen zugeordnet sind.
- Ohne elementarfunktionalen Bezug

Schritt 3: Funktionale Zuordnung der Klassen



Schritt 4: Identifikation der Frameworks und Zuordnung der Superklassen

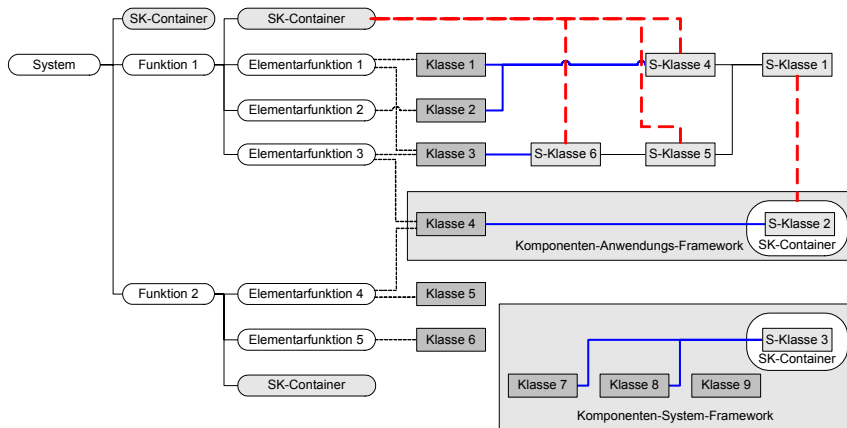
■ Identifikation der Frameworks

- ▶ *Komponenten-System-Framework*: Alle Klassen ohne elementar-funktionalen Bezug
- ▶ *Komponenten- Anwendungs-Framework*: Alle Klassen mit Zuordnung zu mehreren Elementarfunktionen verschiedener Funktionsbereiche

■ Zuordnung der Superklassen

- ▶ Elementarfunktionale Zuordnung der Superklassen
- ▶ Funktionale Zuordnung der Superklassen
- ▶ Überfunktionale Zuordnung der Superklassen

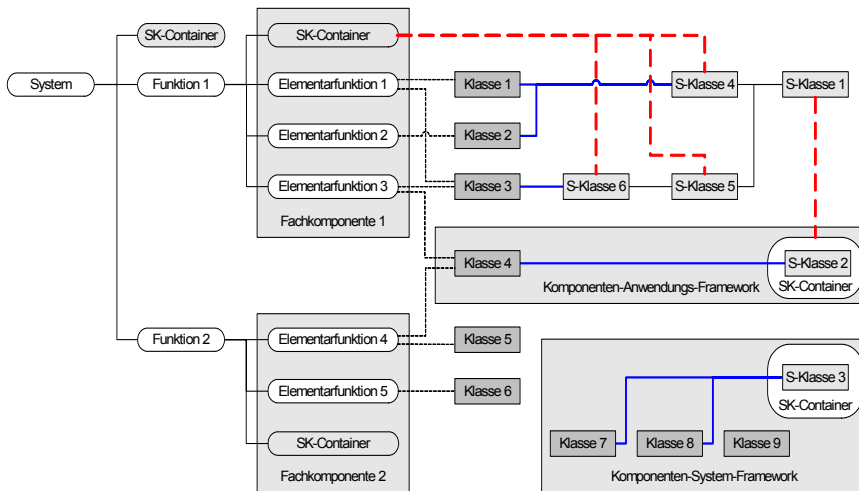
Schritt 4: Identifikation der Frameworks und Zuordnung der Superklassen



Schritt 5: Identifikation der Fachkomponenten

- Eine Fachkomponente setzt sich zusammen aus:
 - ▶ Klassen mit eindeutiger Zuordnung zu einem Funktionsbereich
 - ▶ Nur Klassen enthalten, die zur Erfüllung der einzelnen Elementarfunktionen dieser Komponente verwendet werden.
- ⇒ Ziel ist die Minimierung des Kommunikations- und Koordinationsaufwandes zwischen zwei Fachkomponenten
- Grenzbereiche der Komponentenidentifikation bei AS, deren Funktionen sich aus einer
 - ▶ Extrem hohen oder
 - ▶ Extrem niedrigen
- Anzahl von Elementarfunktionen zusammensetzen.

Schritt 5: Identifikation der Fachkomponenten



Zusammenfassung und erweiterte Einsatzmöglichkeiten

■ Ergebnis

- ▶ Komponentenbasiertes Anwendungssystem
- ▶ „Kleine“ Systemdokumentation

■ Erweiterung

- ▶ Keine Einschränkungen bzgl. der Anwendung
- ▶ Ausweitung auf Schichtenarchitekturen