20 Entwurfsmuster

20.1 Grundsätzliches

Motivation:

- Bereitstellung bewährter, vorgefertigter Lösungsstrukturen für wiederkehrende Entwurfsprobleme
- Schaffung einer begrifflichen Basis und Terminologie für die Kommunikation über solche Probleme

Entwurfsmuster (design pattern) – spezielle Komponente, die eine allgemeine, parametrierbare Lösung für ein typisches Entwurfsproblem bereitstellt.

Grundsätzlich gibt es eine Fülle von Mustern, sowohl allgemeiner Art als auch für spezifische Anwendungsbereiche.

Musterkataloge erschließen das Wissen über Muster und machen die Muster wiederverwendbar.

20.2 Ausgewählte Beispiele für Entwurfsmuster

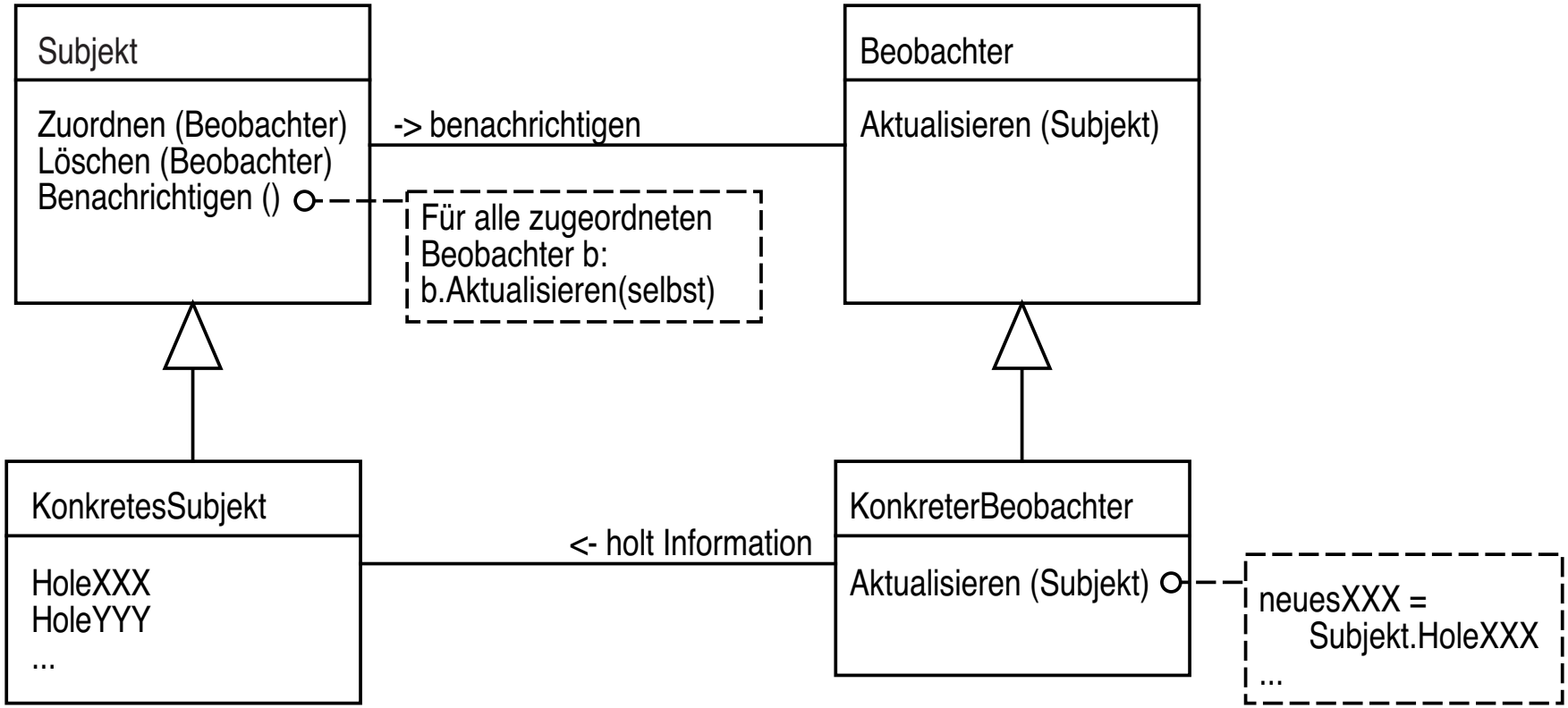
Das Beobachtermuster (observer pattern)

Intention: Objekte können Daten bei einem Informationsanbieter abonnieren. Bei jeder Änderung der subskribierten Daten werden die Abonnenten automatisch über die Änderung informiert bzw. werden ihnen die geänderten Daten zugestellt.

Typische Anwendungen:

- Entkopplung voneinander abhängiger Objekte durch Ersetzung direkter Kommunikation über Aufrufe durch eine indirekte über Benachrichtigung
- Trennung von Informationsbereitsteller und einer Menge von Informationsverarbeitern bzw. Darstellern
- Beispiel eines Verhaltensmusters (behavioral pattern)

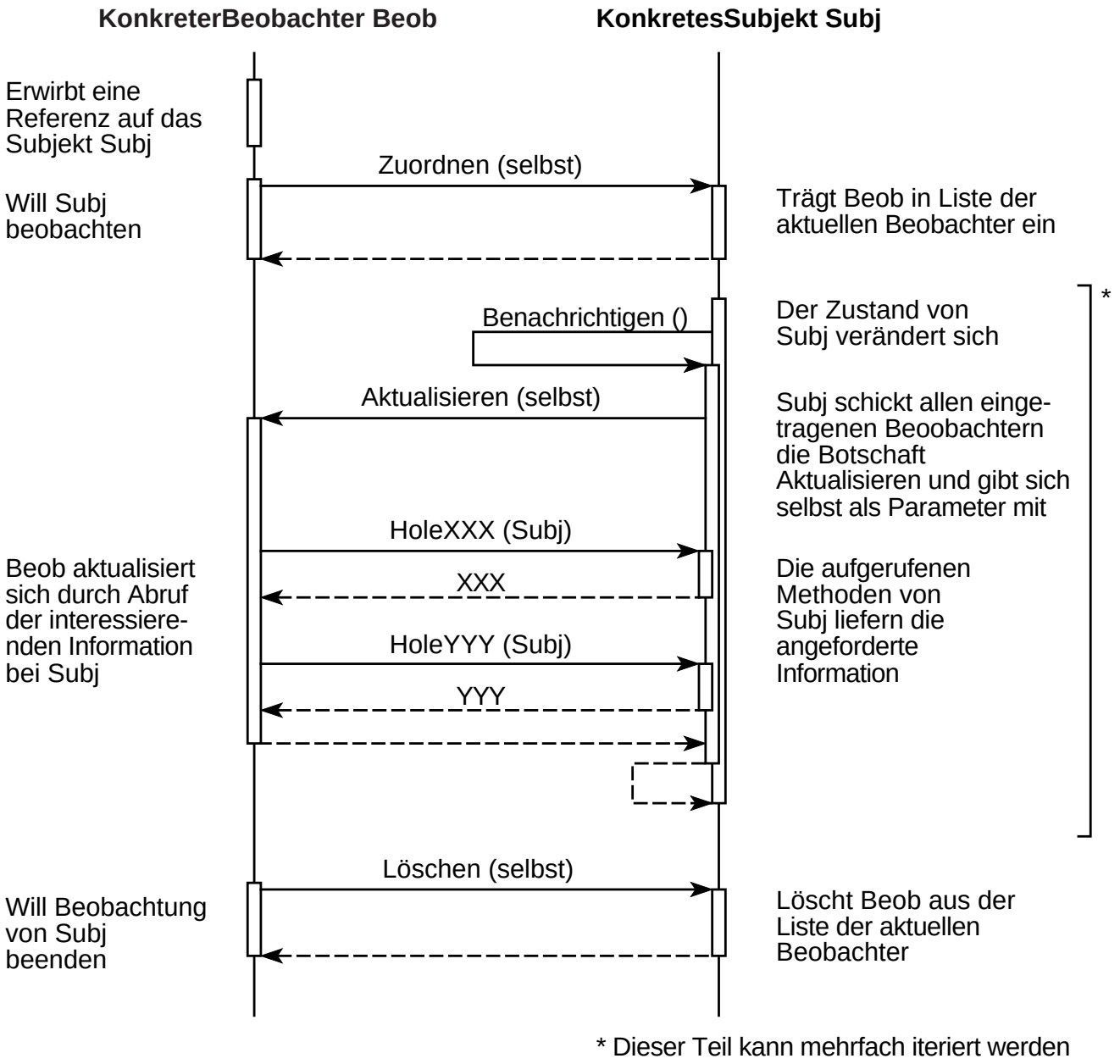
Struktur des Beobachtermusters



Arbeitsweise des Beobachtermusters

- Jedes Objekt der Klasse Subjekt führt eine Liste von Beobachtern, welche an Veränderungen im Zustand dieses Objekts interessiert sind. Zuordnen bzw. Löschen fügt Beobachter in die Liste ein bzw. entfernt sie.
- Nach jeder Veränderung schickt das Gegenstandsobjekt sich selbst die Botschaft Benachrichtigen. Diese iteriert die Liste der Beobachter und schickt jedem Beobachter die Botschaft Aktualisieren. Das benachrichtigende Objekt wird als Parameter mitgegeben. Ggf. können weitere Informationen (z.B. die Art des eingetretenen Ereignisses) als Parameter mitgegeben werden.
- Jeder benachrichtigte Beobachter reagiert, indem er beim benachrichtigenden Gegendstandsobjekt mit HoleXXX-Botschaften die ihn interessierenden Informationen abruft.
- Alternativ können der Update-Botschaft die veränderten Daten gleich mitgegeben werden, wodurch der Abruf durch HoleXXX entfällt. Dieses Bringprinzip ist effizient, koppelt aber Gegenstand und Beobachter stärker als das mit HoleXXX realisierte Holprinzip.

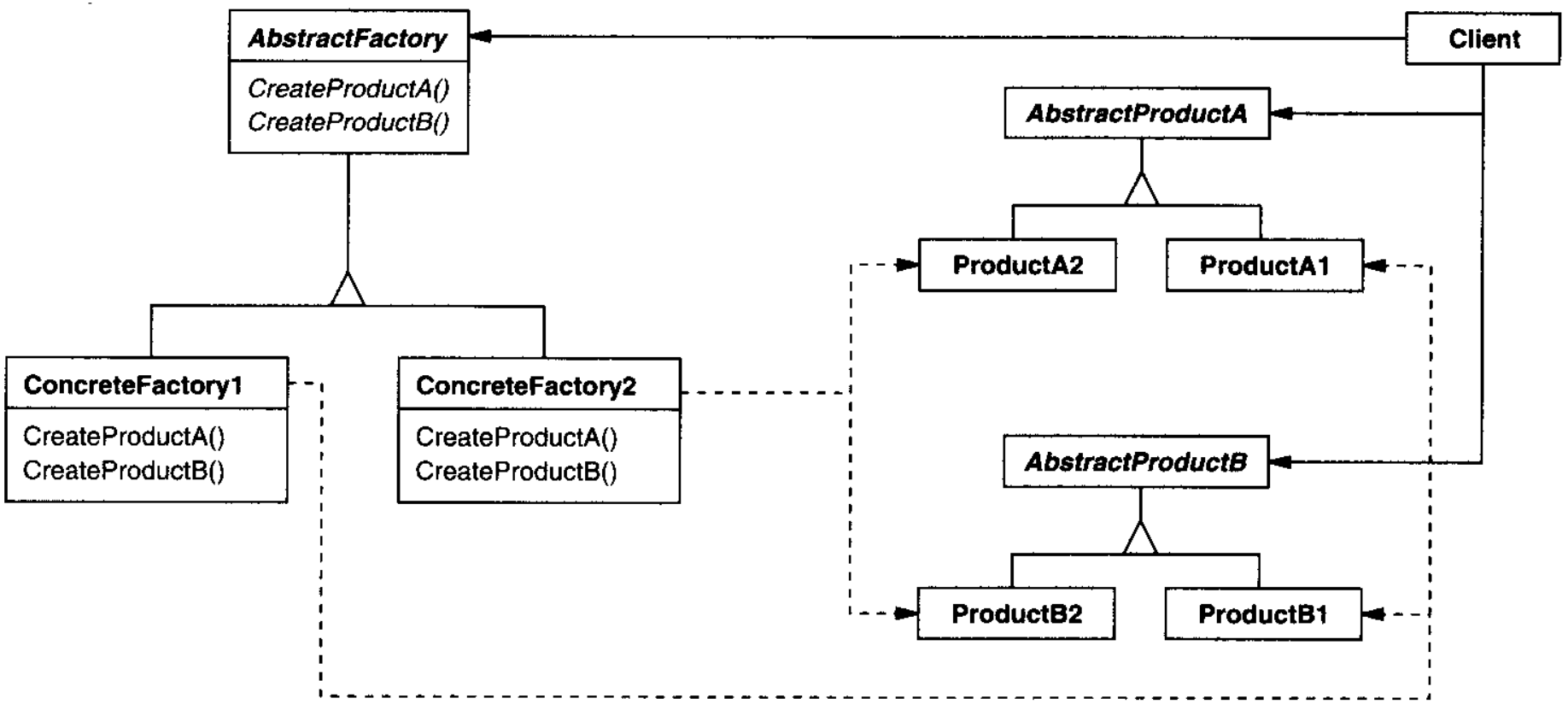
Benutzungsszenario:



Das Fabrikmuster (Abstract factory pattern)

- Beispiel eines Erzeugungsmusters (creational pattern)
- Ermöglicht die Erzeugung von Familien verwandter Objekte, ohne dass das die Namen der Klassen, aus denen das zu erzeugende Familienmitglied besteht, bekannt sein müssen.
- Grundgedanke: Realisierung von Variabilität ohne Fallunterscheidungen in den Programmen (und damit leichtere Erweiterbarkeit)

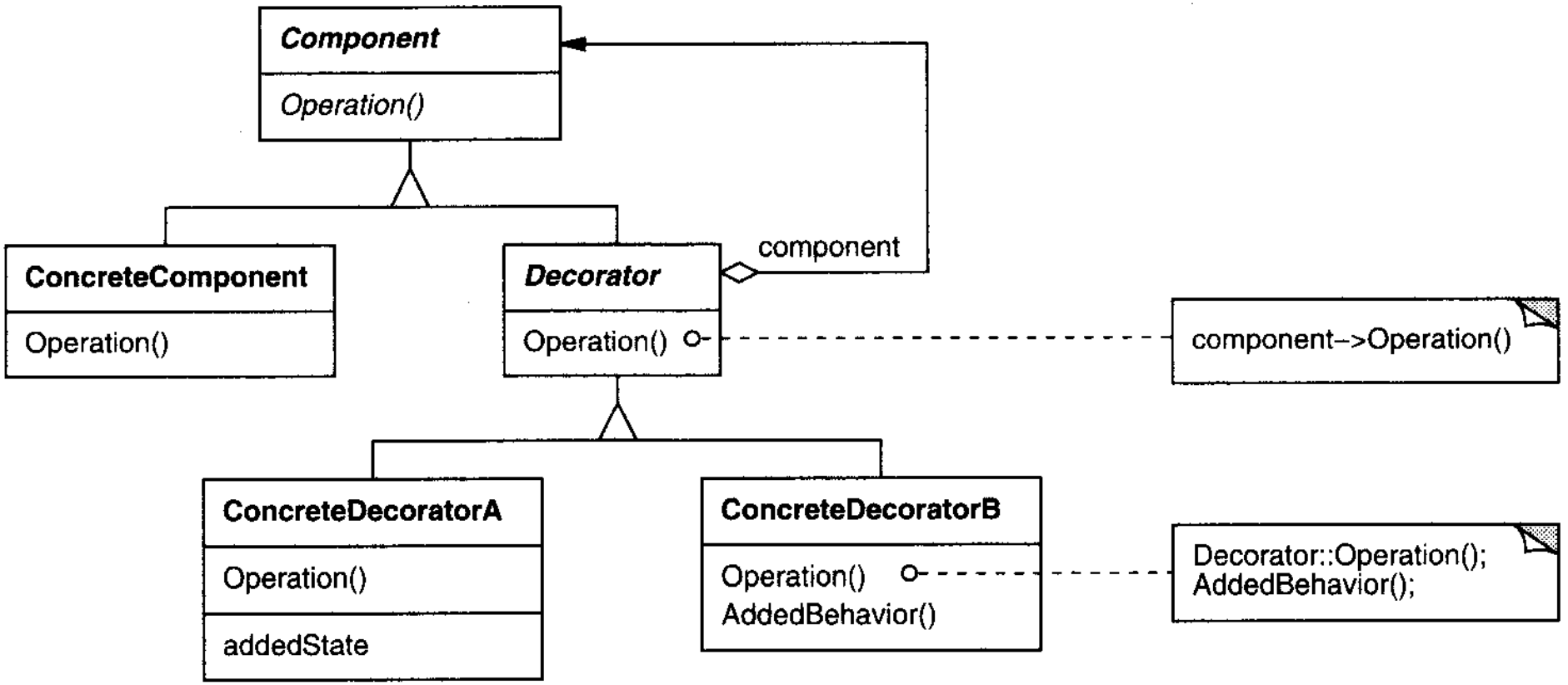
Struktur des Fabrikmusters



Das Dekorierermuster (Decorator pattern)

- Beispiel eines Strukturmusters (structural pattern)
- Erweiterung der Funktionalität eines Objekts durch eine Kette von Methodenaufrufen statt durch Unterklassenbildung
- Im Gegensatz zur Unterklassenbildung kann die Funktionalität eines Objekts zur Laufzeit dynamisch verändert werden

Struktur des Dekorierermusters



20.3 Entwurf mit Mustern

- Voraussetzung: Grundschatz an Mustern kennen (Lernen / Erfahrung / Entwürfe und Code anderer lesen bzw. inspizieren)
- Bei der Modularisierung Anwendungssituationen für Muster erkennen
  ⇨ entsprechende Muster verwenden
- Allgemeine Muster wie Beobachter, Mediator oder Fabrik beschreiben weitgehend reine Lösungsstrukturen und haben keine Entsprechung in Objekten der Aufgabenstellung
- Daneben gibt es auch anwendungsspezifische Muster, welche die Bildung von Aufgabenmodellen erleichtern

20.4 Motivation für Architekturmuster

- Bereitstellung bewährter, vorgefertigter Grundstrukturen für wiederkehrende Architekturprobleme

Architekturmuster – Allgemeine, parametrierbare Architekturschablone für eine typische Problemklasse.

Abgrenzung Stil - Muster - Metapher:

- Architekturmetapher – Leitbild für das Gliedern und Verstehen einer Architektur
- Architekturstil – eine bestimmte Art des Zusammenwirkens von Komponenten und Interaktionen
- Architekturmuster – Allgemeine, parametrierbare Architektur für eine typische Problemstellung

Abgrenzung zum Entwurfsmuster:

- Entwurfsmuster sind Muster für Komponenten, nicht für ganze Architekturen

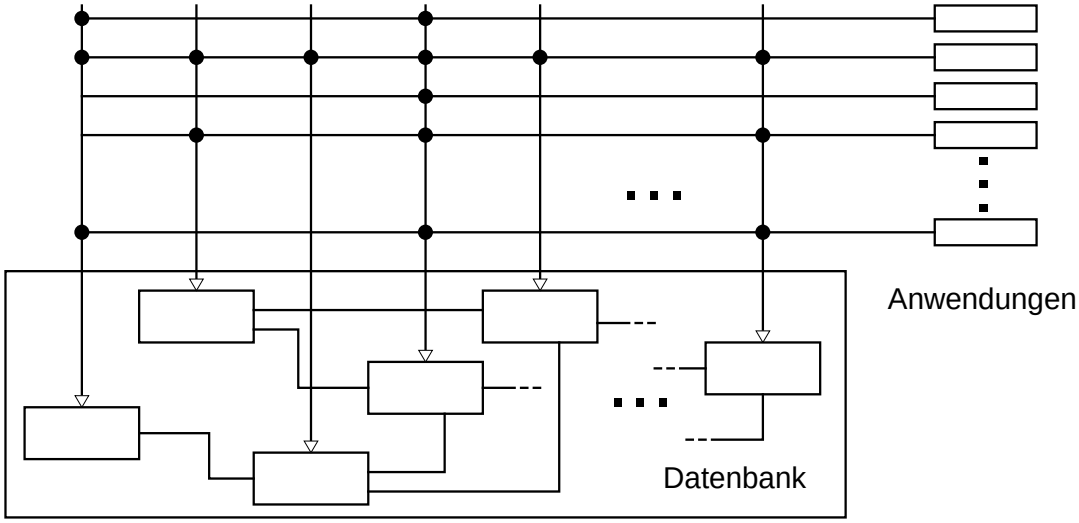
Abgrenzung zum Rahmen (framework):

- Rahmen enthalten fertig codierte Teile, während ein Muster nur eine Konstruktionsschablone darstellt.

20.5 Einige typische Architekturmuster

Strukturmuster; Beispiel: Matrixmuster

- System besteht aus Menge von Daten- und Funktionsmodulen
- Jede Funktion kann auf jedes Datum zugreifen
- Funktionsmodule enthalten keine permanenten Daten
- Muster für die Klassische Architektur datenbankbasierter Systeme

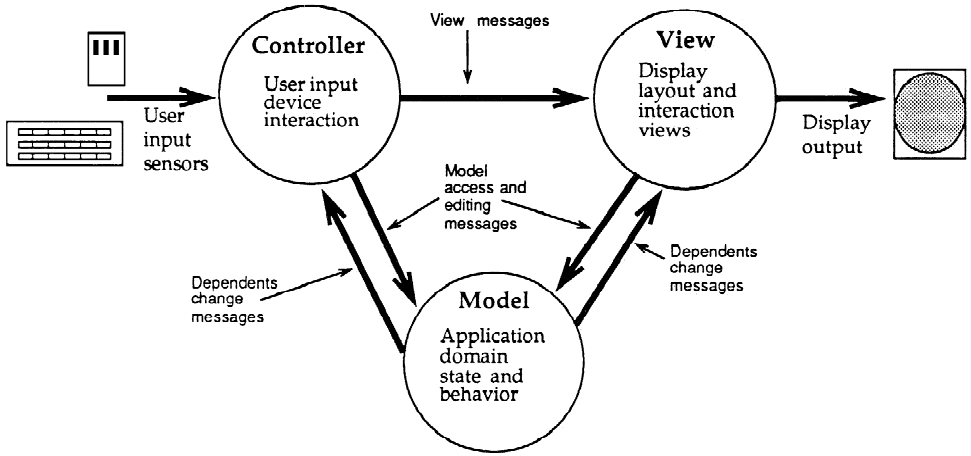


Steuermuster

- EVA (Eingabe-Verarbeitung-Ausgabe)
  Ein Steuermodul steuert nacheinander (in Sequenz oder iterativ) Eingabe-, Verarbeitungs- und Ausgabemodule an.
- Hauptschleife
  Ein Prozess misst, regelt und steuert, indem er in einer Endlosschleife zyklisch alle Datenquellen (Sensoren, etc.) abfragt und alle Datensenken (Anzeigen, Aktuatoren...) mit aktualisierten Werten versorgt.
- Hollywood (“Don’t call us, we call you”)
  Ein Ereignisverwalter registriert alle Eingabeereignisse und ruft die zugehörigen Dienste auf. Das Anwendungsprogramm enthält kein Hauptprogramm mehr.

Modularisierungs-/Entkopplungsmuster

- Benutzungshierarchie
  Ein System ist strikt nach dem Delegationsparadigma organisiert. Die Benutzungsbeziehungen bilden einen gerichteten, azyklischen Graph.
- Model-View-Controller (MVC)
  Gliederung eines Systems in ein “Model” (Anwendungslogik, Modell des Anwendungsbereichs), eine “View” (äußere, sichtbare Repräsentation) und einen “Controller” (Behandlung aller Benutzereingaben)



Aus: Krasner und Pope 1988

Verteilungsmuster

- Client /Server
  siehe Kapitel 23
- TP-Monitor
  Middleware-Architektur für datenbankbasierte Systeme mit einem Transaktionsverwalter als Hauptkomponente
- Three-Tier
  Middleware-Architektur für datenbankbasierte Systeme, bei der die Middleware einen Teil der Anwendungslogik enthält
- Komponentenbus
  siehe Kapitel 23