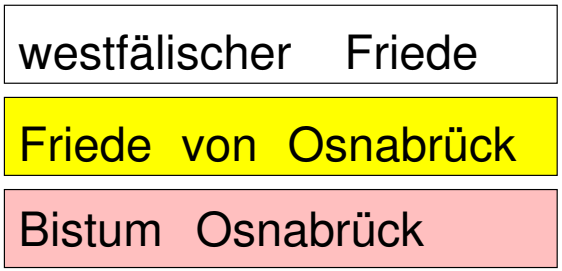
Bewertung von Modularisierung

Software Architecture Analysis Method (SAAM) (Kazman u. a. 1996):

1. Lege wichtige Qualitätsaspekte fest
2. Beschreibe alternative Modularisierungen
3. Entwickle Szenarien für die Qualitätsaspekte
4. Spiele die Szenarien durch und bewerte die Modularisierungen
5. Betrachte Wechselwirkungen zwischen den Qualitätsaspekten
6. Fasse die Evaluation zusammen

Beispiel: Key Word in Context (KWIC) (Parnas 1972)



Beispiel: Key Word in Context (KWIC) (Parnas 1972)

zyklische Vertauschung
jeder Zeile

westfälischer Friede

Friede von Osnabrück

Bistum Osnabrück

westfälischer Friede

Friede westfälischer

Friede von Osnabrück

Osnabrück Friede von

von Osnabrück Friede

Bistum Osnabrück

Osnabrück Bistum

Beispiel: Key Word in Context (KWIC) (Parnas 1972)

zyklische Vertauschung jeder Zeile

Sortierung der Zeilen

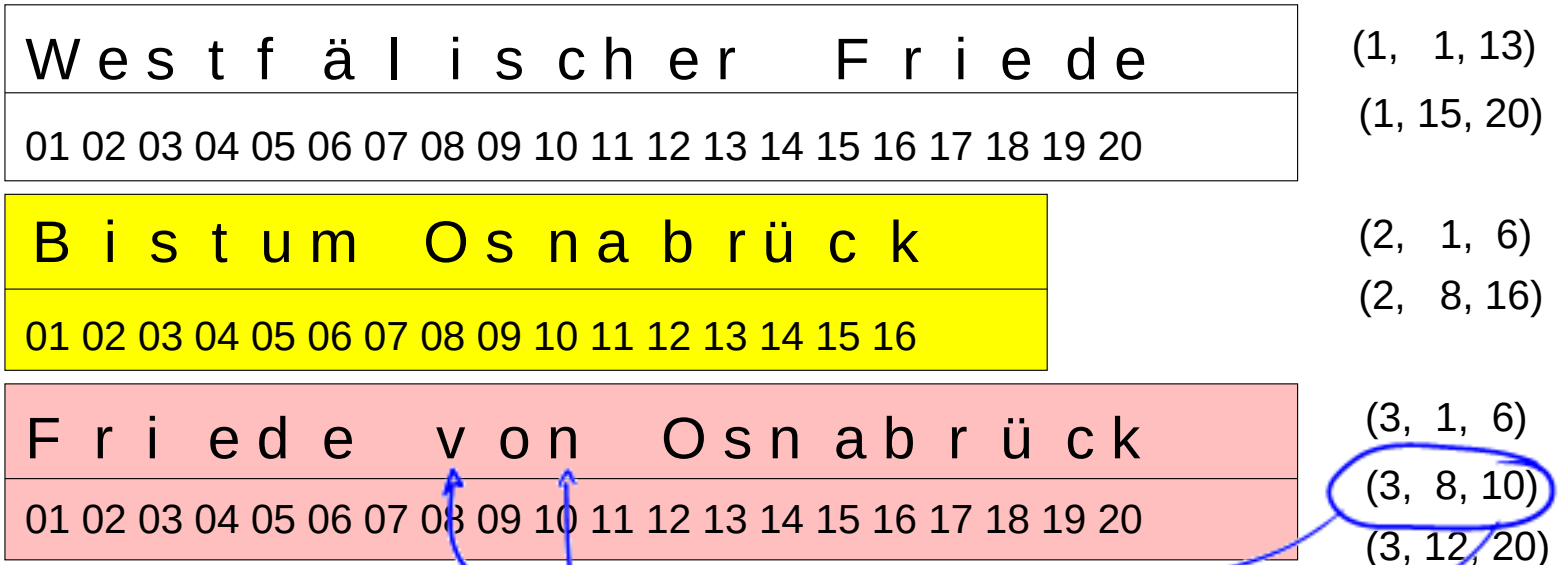
westfälischer Friede	westfälischer Friede	Bistum Osnabrück
Friede von Osnabrück	Friede westfälischer	Friede von Osnabrück
Bistum Osnabrück	Friede von Osnabrück	Friede westfälischer
	Osnabrück Friede von	Osnabrück Bistum
	von Osnabrück Friede	Osnabrück Friede von
	Bistum Osnabrück	von Osnabrück Friede
	Osnabrück Bistum	westfälischer Friede

Betrachtete Qualitätsaspekte und Szenarien

- Änderbarkeit
  - Eingabeformat ändert sich
  - größere Dateien müssen verarbeitet werden
  - riesige Dateien müssen verarbeitet werden
- separate Entwickelbarkeit
  - verteile Arbeit an Entwicklungsteams
- Performanz
  - berechne KWIC für FB3-Webseiten

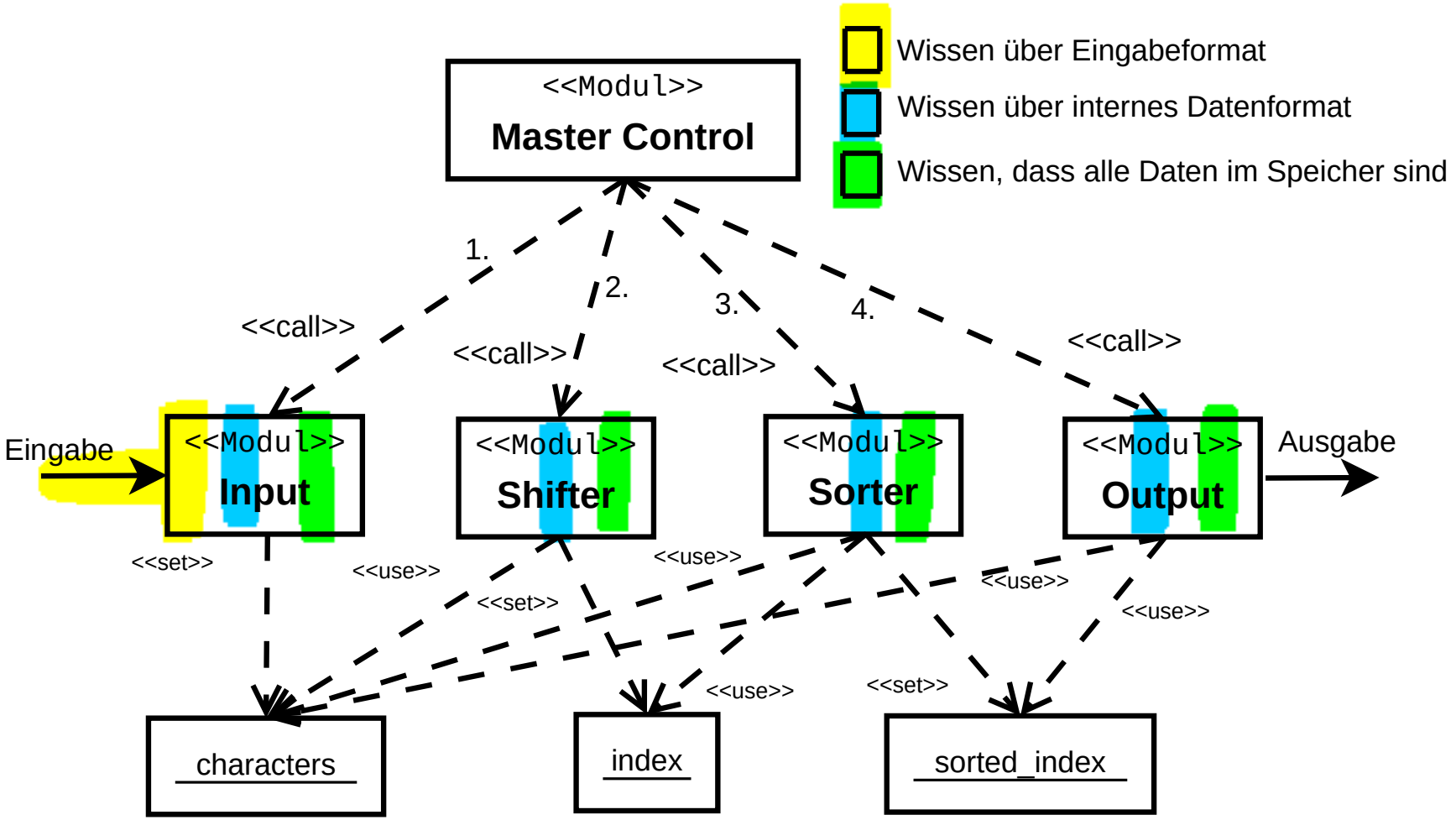
Ablauf

1. Eingabe der Daten
2. Interne Speicherung der Daten
3. Indizierung (Zeile, Wortanfang, Wortende)



4. Zyklische Rotation der Indizierung
5. Sortierung der Indizierung
6. Ausgabe der Indizierung

Erste Modularisierung (ablauforientiert)



Qualitäten

Änderbarkeit:

	betroffene Module			
	Input	Shifter	Sorter	Output
Eingabeformat	×			
größere Dateien	×	×	×	×
riesige Dateien	×	×	×	×

Getrennte Entwicklung:

- alle Datenstrukturen müssen bekannt sein
- komplexe Beschreibung notwendig

Performanz:

- schneller Zugriff auf globale Variablen

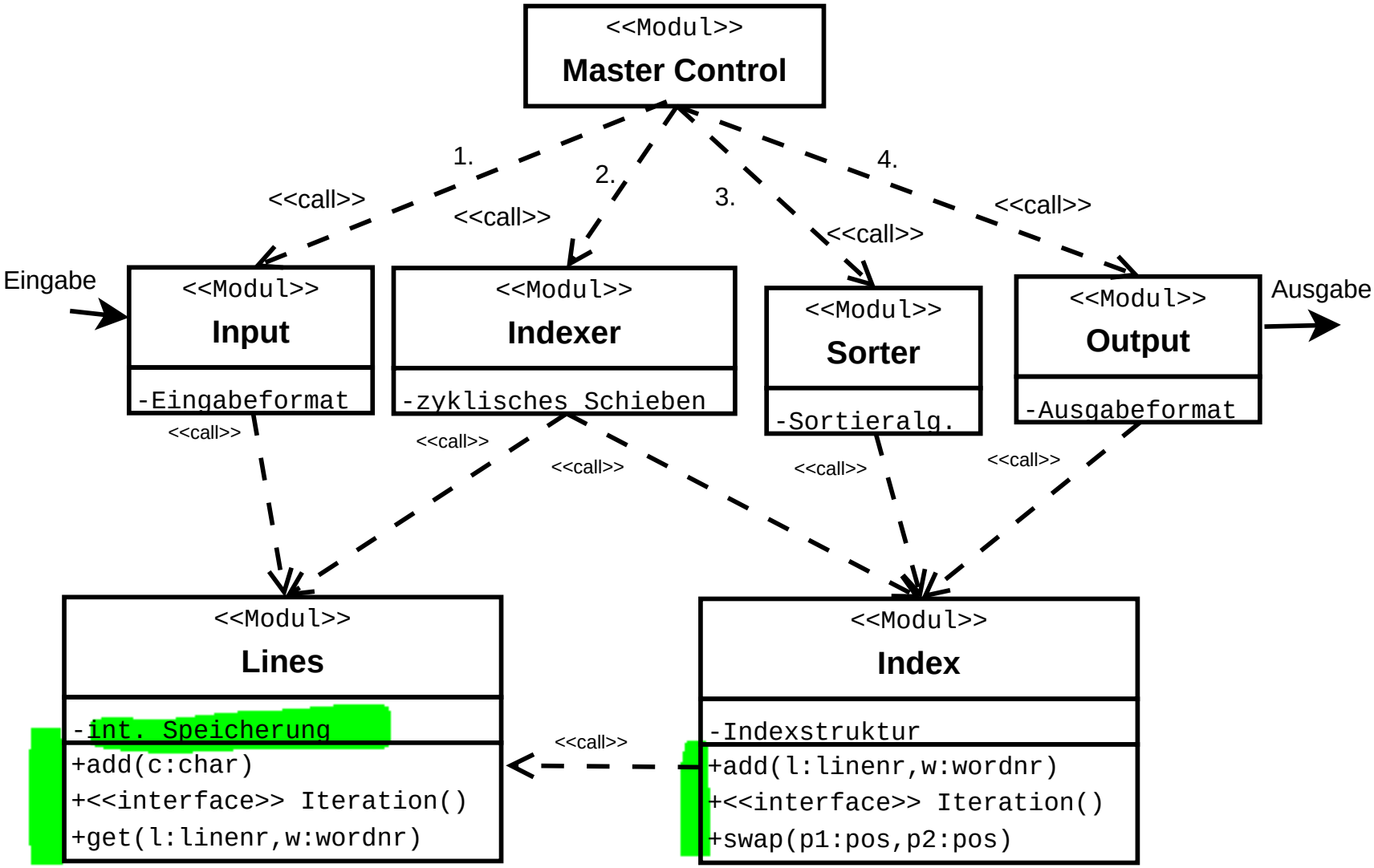
Geheimnisprinzip (Information Hiding) (Parnas 1972)

Definition

Geheimnisprinzip: Module verbergen alle Implementierungsentscheidungen, die sich ändern können, hinter einer abstrakten Schnittstelle.

Zweite Modularisierung (objektbasiert)

. . . nach dem Geheimnisprinzip (Information Hiding) (Parnas 1972)



Qualitäten

Änderbarkeit:

	betroffene Module					
	Input	Lines	Index	Indexer	Sorter	Output
Eingabeformat	×					
größere Dateien		×				
riesige Dateien		×				

Getrennte Entwicklung:

- nur Schnittstellen müssen bekannt sein

Performanz:

- zusätzliche Aufrufe

Abschließendes Wort zur Modularisierung

Es gibt viele Modularisierungsmöglichkeiten.

Jede ist gut für einen Zweck, schlecht für einen anderen.

Mit gängigen Programmiersprachen muss man sich auf eine festlegen.

Abschließendes Wort zur Modularisierung

Es gibt viele Modularisierungsmöglichkeiten.

Jede ist gut für einen Zweck, schlecht für einen anderen.

Mit gängigen Programmiersprachen muss man sich auf eine festlegen.

Definition

Querschnittsbelange (Cross-Cutting Concerns):
Implementierungsaspekte, die eine große Zahl von Modulen betreffen.

Beispiele: Fehlerbehandlung, Logging-Mechanismen, Vermeidung von Speicherlöchern etc.

Abschließendes Wort zur Modularisierung

Es gibt viele Modularisierungsmöglichkeiten.

Jede ist gut für einen Zweck, schlecht für einen anderen.

Mit gängigen Programmiersprachen muss man sich auf eine festlegen.

Definition

Querschnittsbelange (Cross-Cutting Concerns):
Implementierungsaspekte, die eine große Zahl von Modulen betreffen.

Beispiele: Fehlerbehandlung, Logging-Mechanismen, Vermeidung von Speicherlöchern etc.

Aspektorientierte Programmiersprachen erlauben eine separate Beschreibung dieser Aspekte und ein „Einweben“ des Aspekts in das System.

AspectJ

Architekturstil

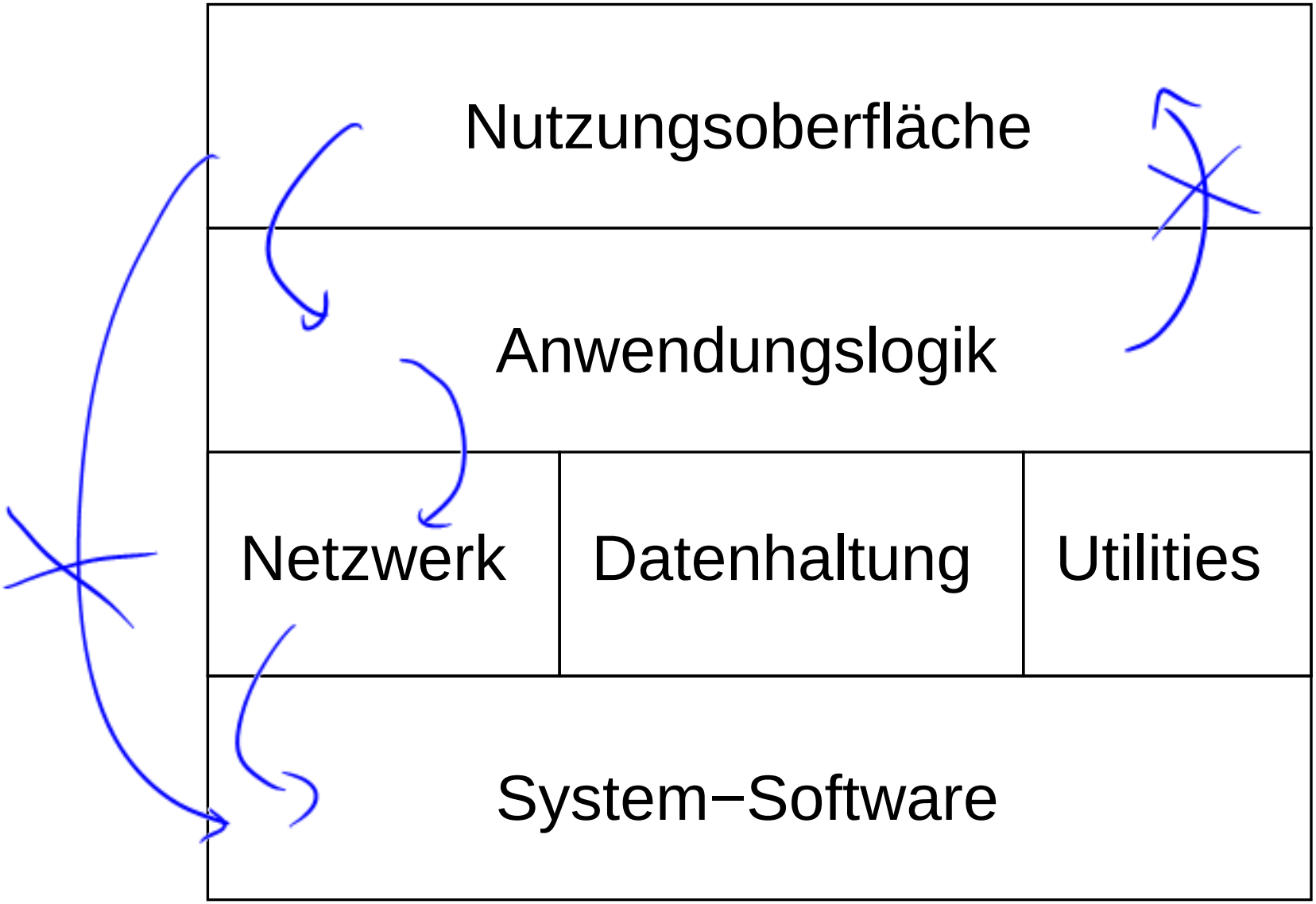
Definition

Architekturstil: beschreibt eine Familie von Architekturen/Systeme als ein Muster der strukturellen Organisation durch

- ein Vokabular (Komponenten- und Konnektorentypen)
- und eine Menge von Einschränkungen, wie Komponenten und Konnektoren verbunden werden dürfen.

Synonyme: Architekturmuster oder Architekturidiom.

Architekturstil: Schichtung



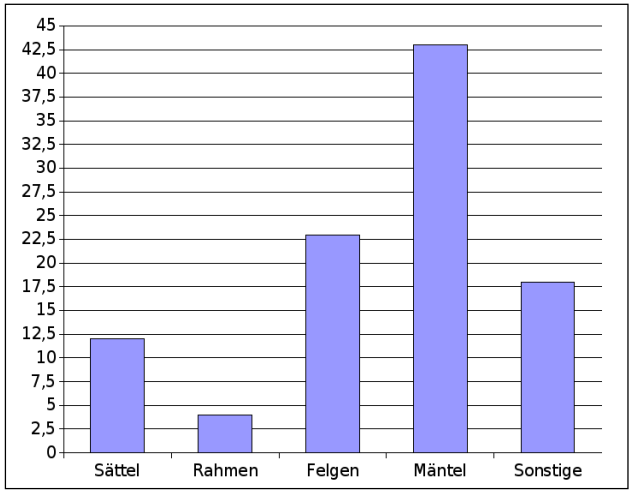
Architekturstil: Schichtung I

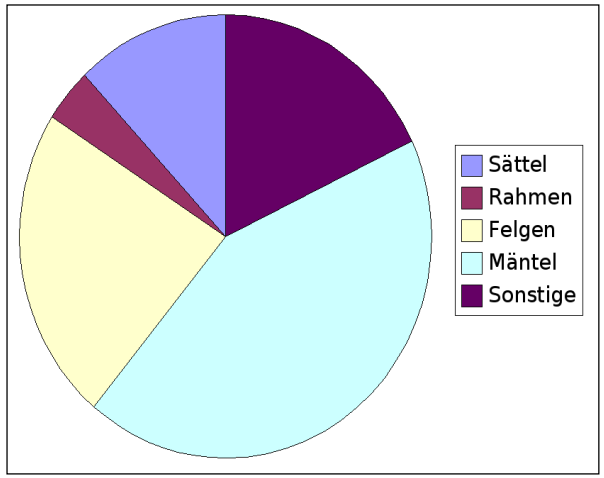
- Vokabular:
  - Komponenten: Module und Schichten
  - Konnektoren: Use-Beziehung
- Struktur:
  - Module sind eindeutig einer Schicht zugeordnet
  - Module einer Schicht dürfen nur auf Module derselben und der direkt darunter liegenden Schicht zugreifen
- Ausführungsmodell:
  - Aufruf von Methoden tieferer Schichten
  - Datenfluss in beide Richtungen (von der unteren Schicht zur oberen durch Rückgabeparameter)

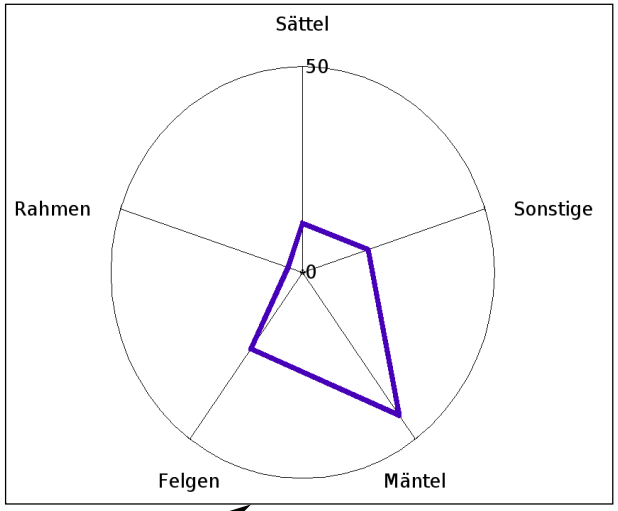
Architekturstil: Schichtung II

- Vorteile:
  - Schicht implementiert virtuelle Maschine, deren Implementierung leicht ausgetauscht werden kann, ohne dass höhere Schichten geändert werden müssen
- Nachteile:
  - höherer Aufwand durch das "Durchreichen" von Information
  - Redundanz durch Dienste tieferer Schichten, die in hohen Schichten benutzt und auf allen Ebenen dazwischen repliziert werden

Anforderungen

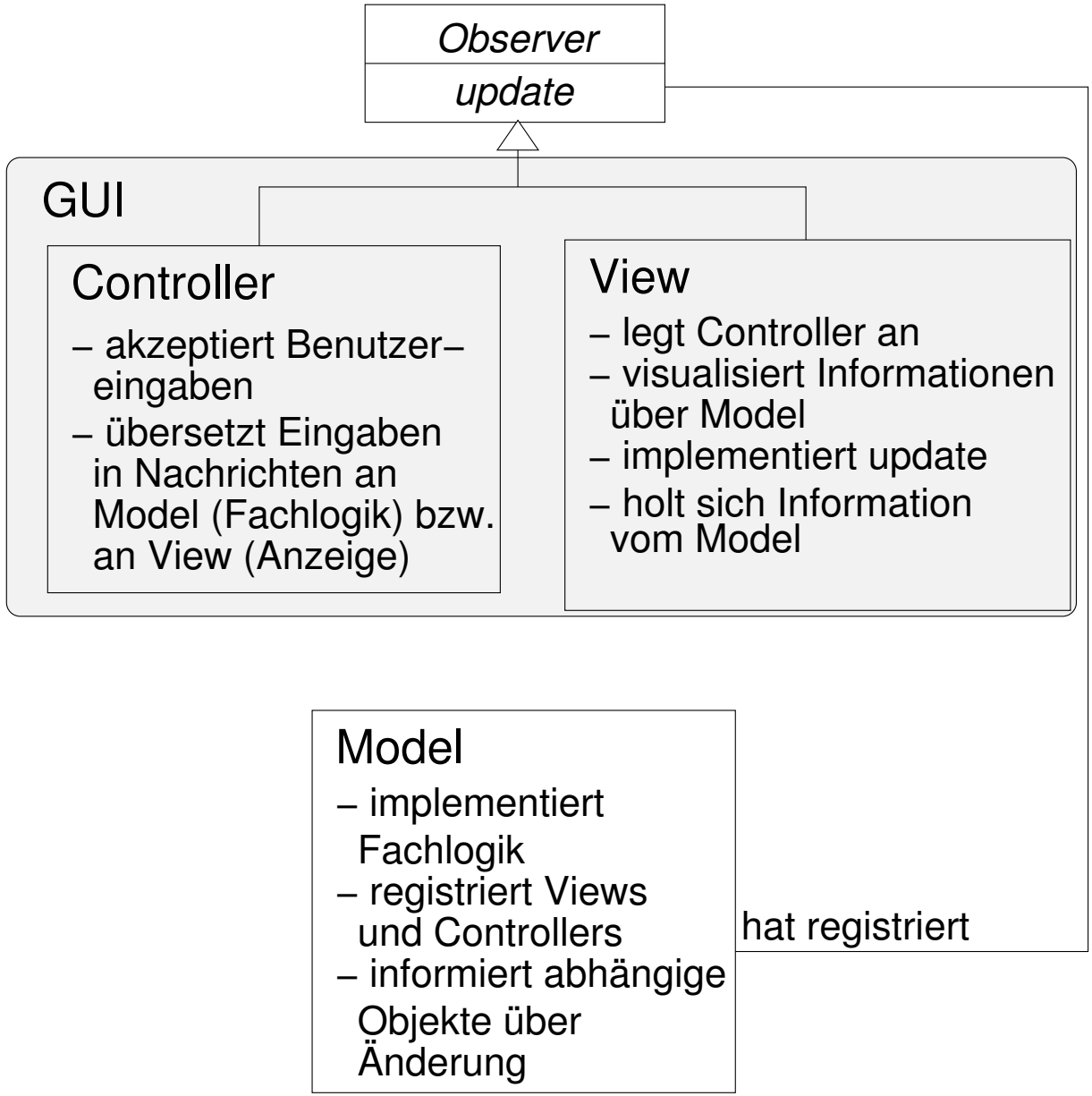




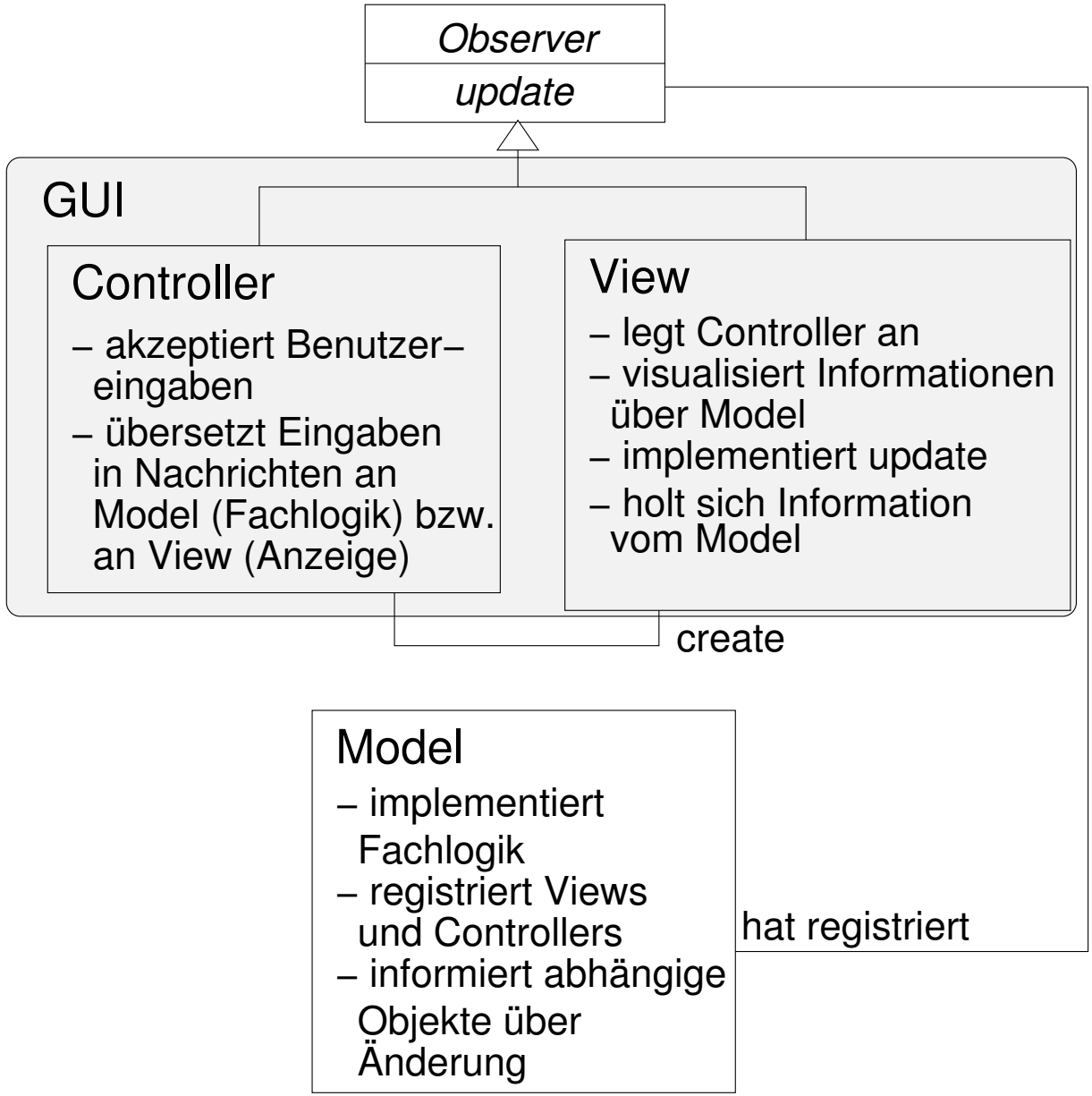


Sättel: 12%
Rahmen: 4%
Felgen: 23%
Mäntel: 43%
Sonstige: 18%

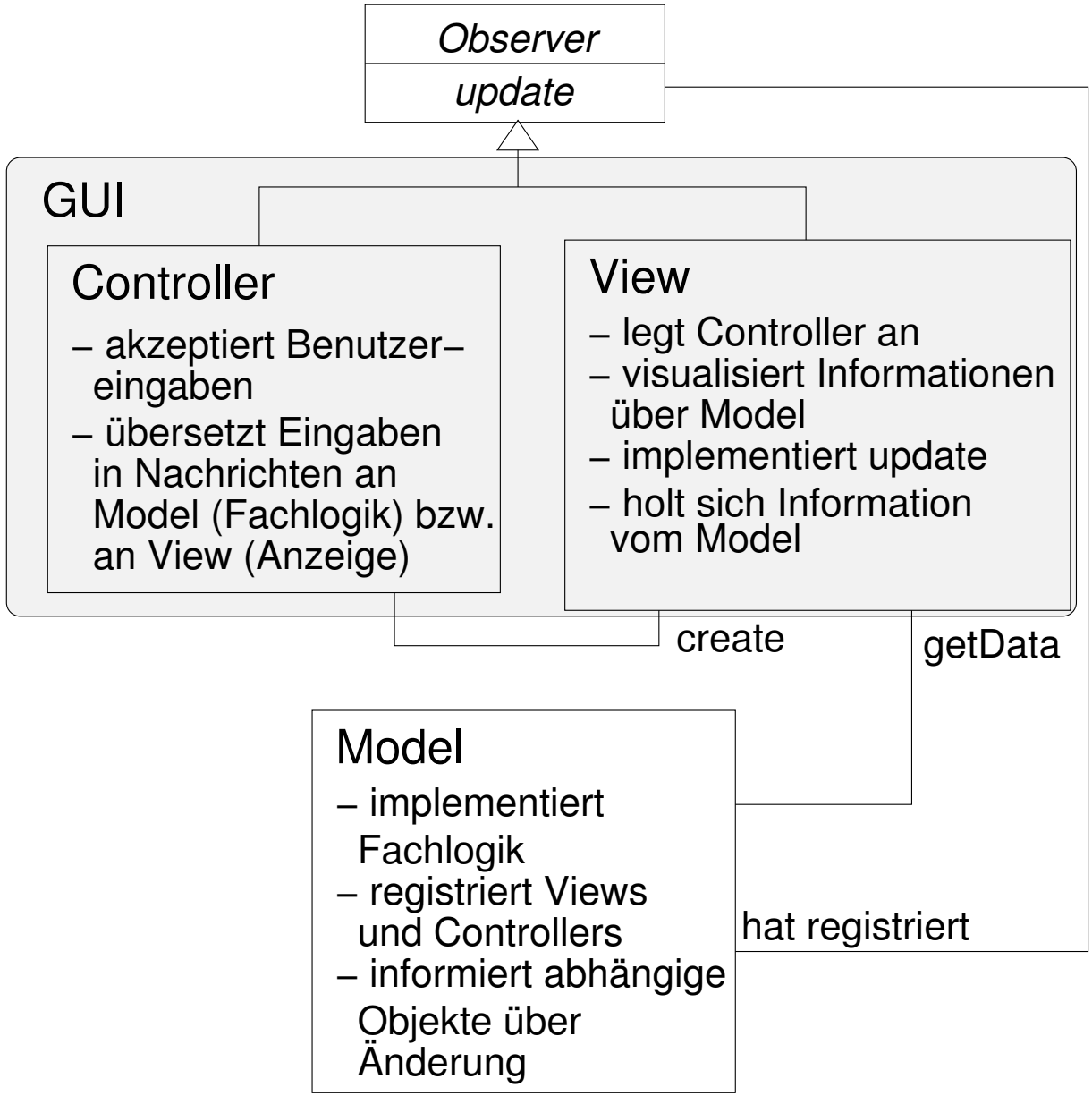
Model-View-Controller (Buschmann u. a. 1996)



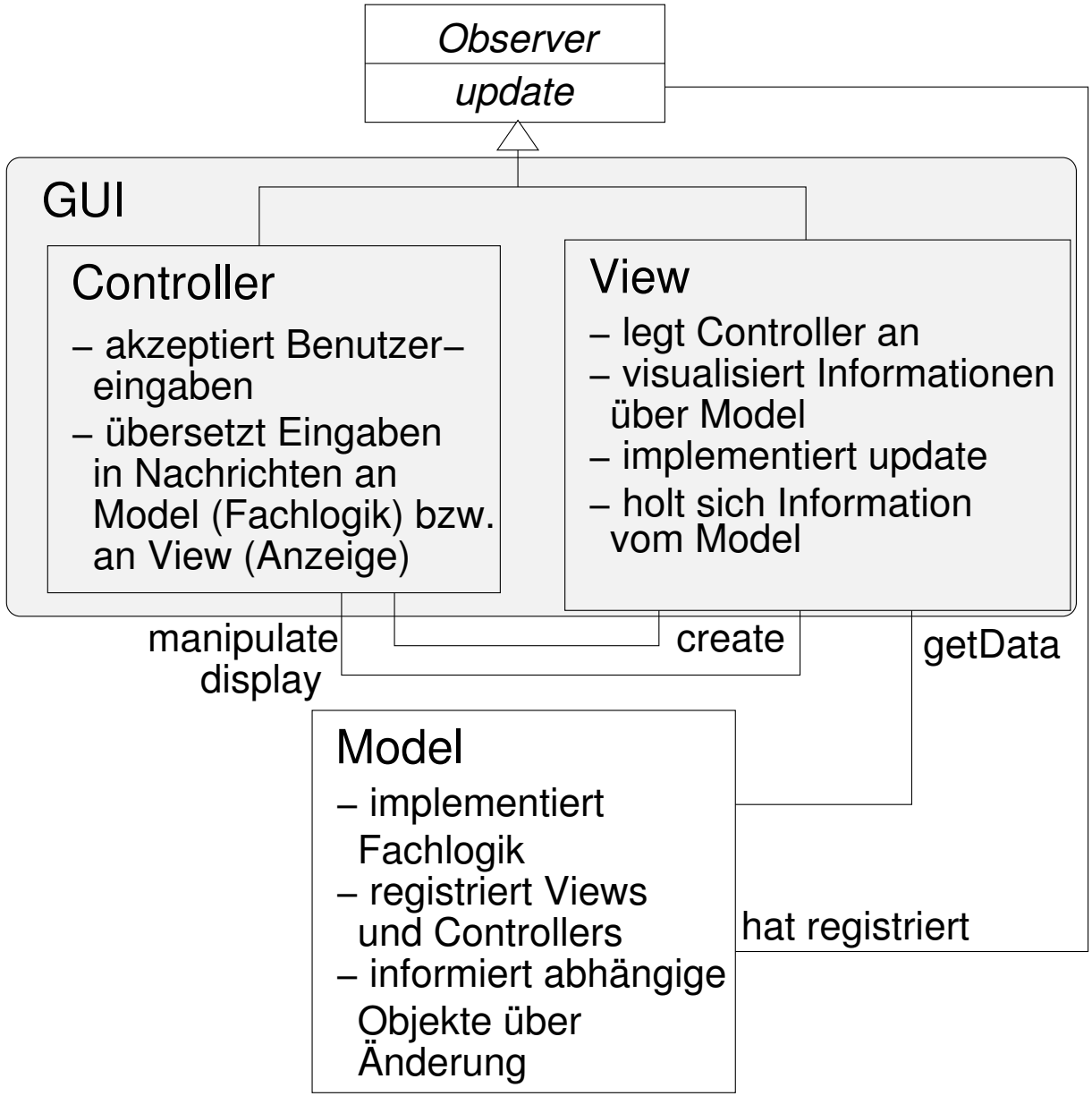
Model-View-Controller (Buschmann u. a. 1996)



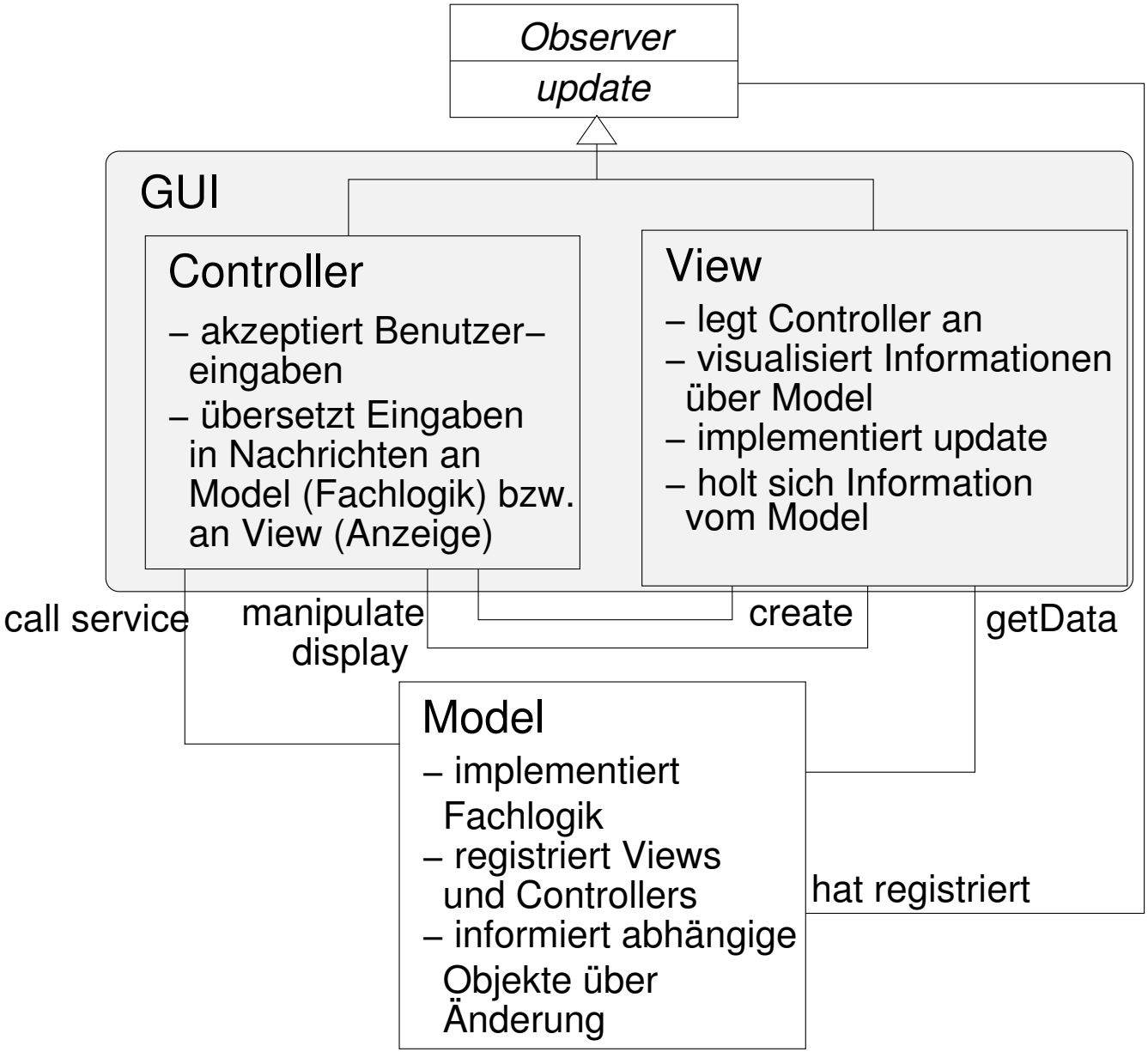
Model-View-Controller (Buschmann u. a. 1996)



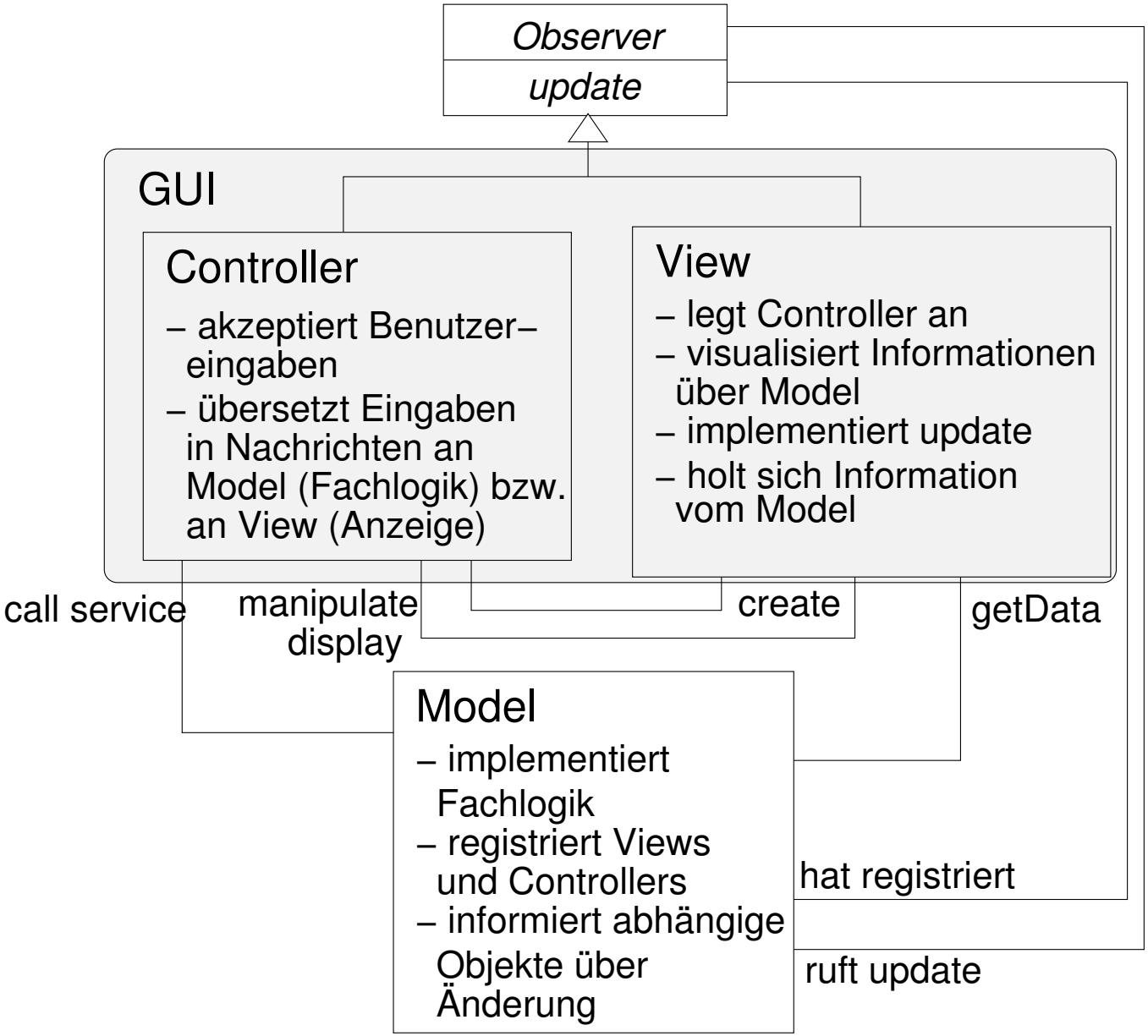
Model-View-Controller (Buschmann u. a. 1996)



Model-View-Controller (Buschmann u. a. 1996)



Model-View-Controller (Buschmann u. a. 1996)



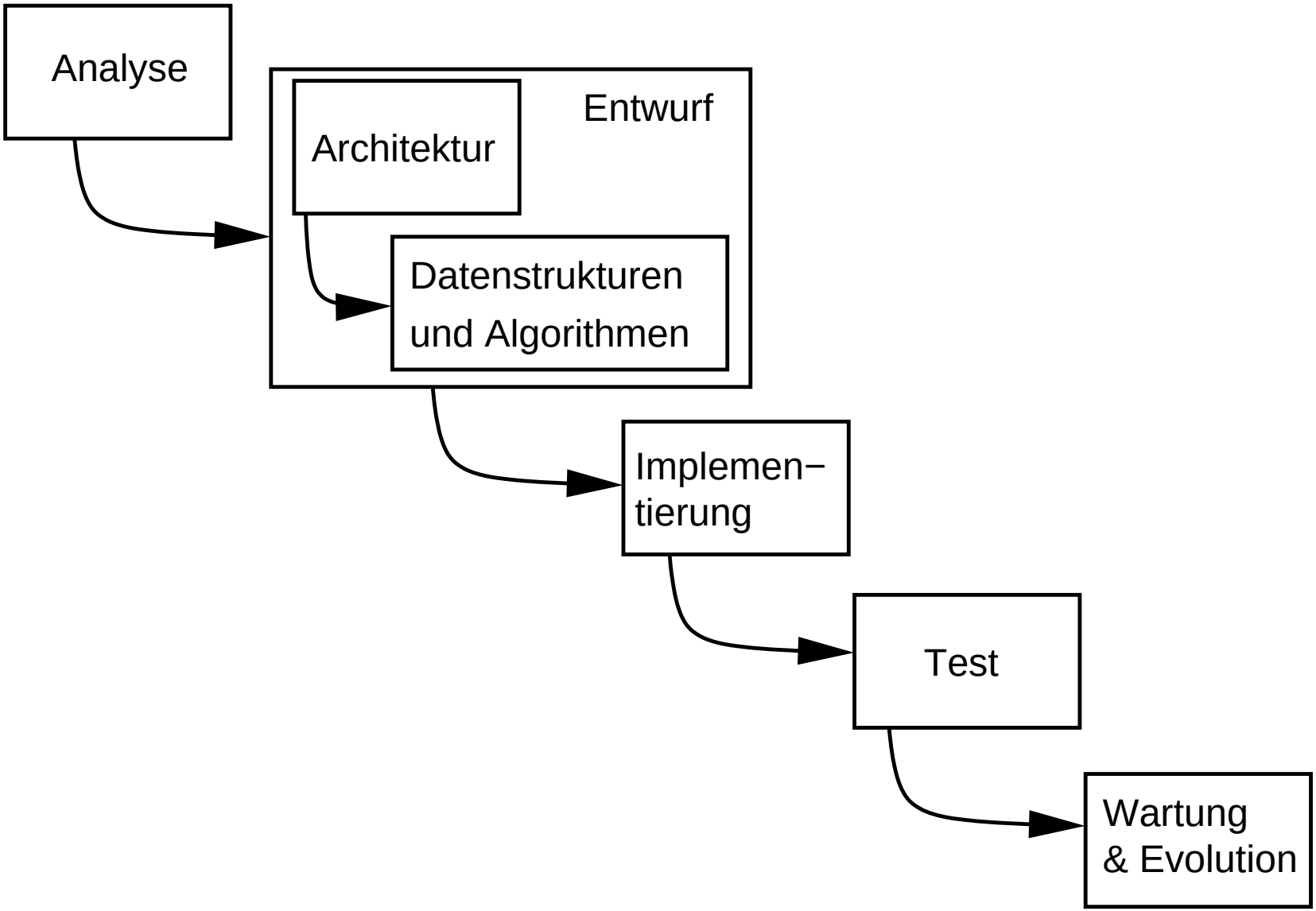
Entwurfsmuster

1. Entwurfsmuster
   - Was ist ein Entwurfsmuster?
   - Bestandteile eines Entwurfsmusters
   - Kategorien von Entwurfsmustern
   - Entwurfsmuster Factory Method
   - Entwurfsmuster Observer

Lernziele

- Verstehen, was Entwurfsmuster sind
- Qualitäten und Einsetzbarkeit der Entwurfsmuster kennen

Kontext


Analyse
Entwurf
Architektur
Datenstrukturen
und Algorithmen
Implemen−
tierung
Test
Wartung
& Evolution

Entwurfsmuster

Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

Christopher Alexander (Architekt und Mathematiker),
"A pattern language", 1977.

Entwurfsmuster

Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

Christopher Alexander (Architekt und Mathematiker),
"A pattern language", 1977.

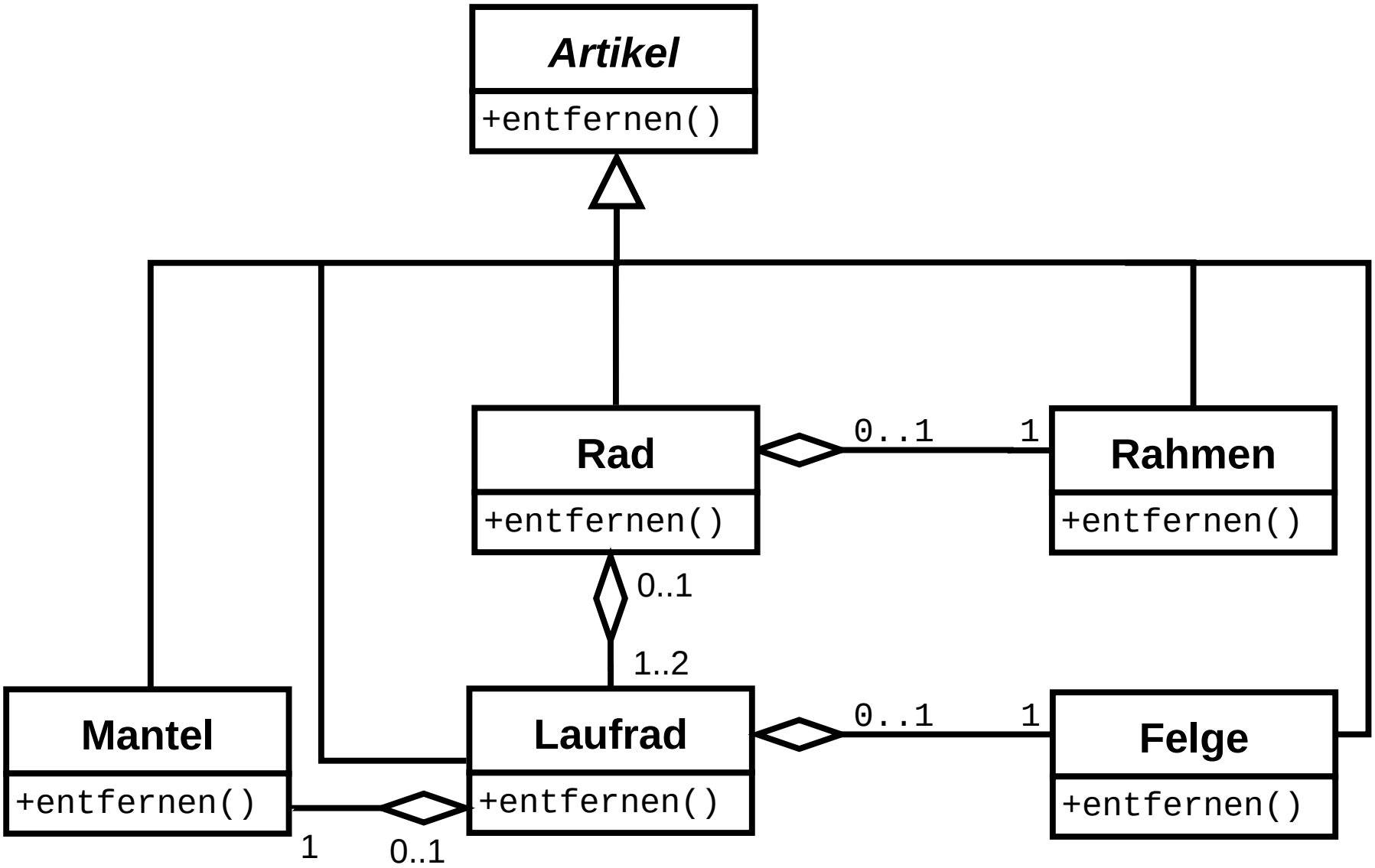
Definition

Entwurfsmuster: „Musterlösung" für ein wiederkehrendes Entwurfsproblem.

Bestandteile eines Entwurfsmusters

- **Name** (kurz und beschreibend)
- **Problem**: Was das das Entwurfsmuster löst
- **Lösung**: Wie es das Problem löst
- **Konsequenzen**: Folgen und Kompromisse des Musters.

Beispielentwurfsproblem



Beschreibung von Mustern (Gamma u. a. 2003) I

- **Name**: *Composite*
- **Zweck**: Teil-von-Hierarchie mit einheitlicher Schnittstelle beschreiben (überall wo ein Ganzes benutzt werden kann, kann auch ein Teil benutzt werden und umgekehrt)
- **Motivation**: . . . Einführung anhand eines konkreten Beispiels. . .
- **Anwendbarkeit**:
  - wenn Teil-von-Beziehung beschrieben werden soll
  - uniforme Schnittstelle für alle Elemente der Hierarchie