25 Architektursprachen

Ausdrucksformen zur Darstellung von Architekturen

- ☆ Natürliche Sprache + informelle Skizzen
- ☆ Teilformale Diagrammsprachen, z. B. UML
- ☆ Formale, auf Architekturprobleme zugeschnittene Sprachen

25.1 Natürliche Sprache + informelle Skizzen

- Ausdrucksmächtig
- Flexibel
- Keine formalen Prüfmöglichkeiten
- Generierung von Coderahmen oder Prototypen nicht möglich

25.2 Teilformale Diagrammsprachen

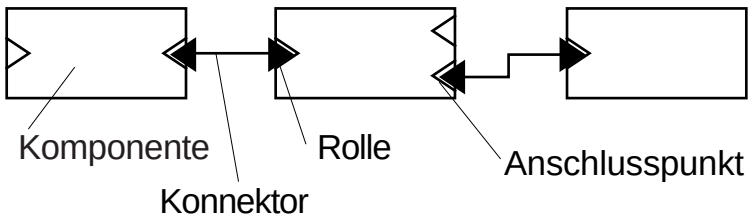
- Ausdrucksmächtig
- Syntaktische Prüfmöglichkeiten
- Codegenerierung in beschränktem Umfang möglich
- Wichtigster Vertreter heute: UML
  - Verschiedene Strukturdiagramme
  - Definierte Syntax
  - Pragmatische Interpretation; weitgehend undefinierte Semantik
  - Vielfalt von Diagrammtypen

25.3 Formale Architektursprachen

- Formal definierte Syntax und Semantik
- Generierung von Code bzw. von Prototypen möglich
- Arbeitet typisch mit typisierten Komponenten und Konnektoren
- Beispiele: Wright, Darwin

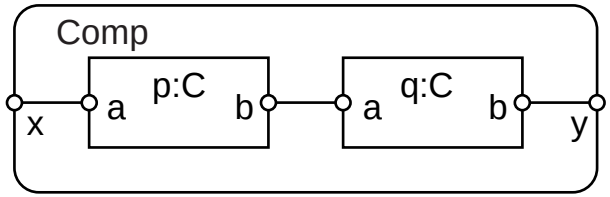
25.3.1 Wright (Garlan und Allen)

- Komponenten (components): verarbeitende Elemente
- Konnektoren (connectors): Verbindungselemente
- Komponenten haben Anschlusspunkte (ports) für Konnektoren
- Konnektoren verbinden Komponenten untereinander
- Konnektoren haben Rollen (roles), die zu den Anschlusspunkten, an die sie andocken, passen müssen
- Anschlusspunkte und Rollen haben explizit definierbare Typen
- Konfigurationen sind Netze aus Komponenten und Konnektoren



25.3.2 Darwin (Kramer und Magee)

- Komponenten sind die verarbeitenden Elemente
- Komponenten haben Schnittstellen
- Schnittstellen verschiedener Komponenten können gebunden werden
- Konfigurationen sind hierarchisch strukturierte Netze von Komponenten, deren Schnittstellen aneinander gebunden sind
- Keine explizite Modellierung von Konnektoren




```
component Comp {
portal x; y;
inst p: C; q: C;
bind  p.a -- x
      q.b -- y
      p.b -- q.a
}
```