PURE/OSEK – Eine aspektorientierte Betriebssystemfamilie für Kraftfahrzeuge

Olaf Spinczyk und Wolfgang Schröder-Preikschat
{os,wosch}@informatik.uni-erlangen.de
Universität Erlangen-Nürnberg, Informatik 4

pure-systems

Danilo Beuche und Holger Papajewski
{danilo.beuche,holger.papajewski}@pure-systems.com
pure-systems GmbH

Überblick

- Motivation
- Ansatz: Programmfamilien im Betriebssystembau
- PURE/OSEK
- Modelle und Werkzeuge
  - Merkmalmodelle → PURE::Consul
  - AOP → AspectC++
- Ergebnisse

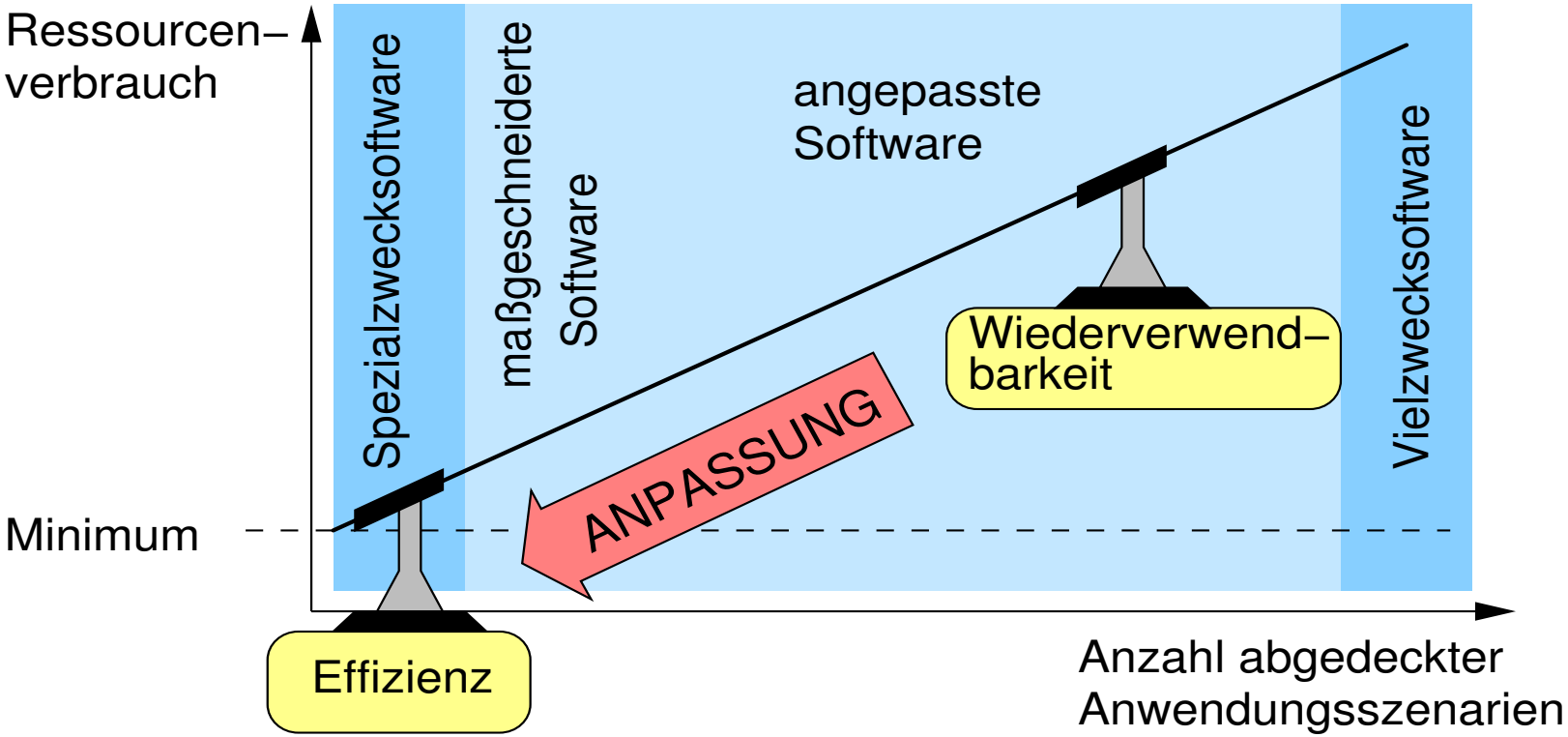
Motivation

- Erfahrungen mit Betriebssystemanwendern:
  - „Isolation kostet zu viele Taktzyklen“
  - „der generierte Code benötigt lediglich einen Thread“
  - „wozu brauche ich überhaupt ...?“
- ➔ Es gibt sehr unterschiedliche Anforderungen
- ➔ Vielseitigkeit wird gelegentlich zur Last:

  unbenötigte Features verbrauchen Ressourcen!

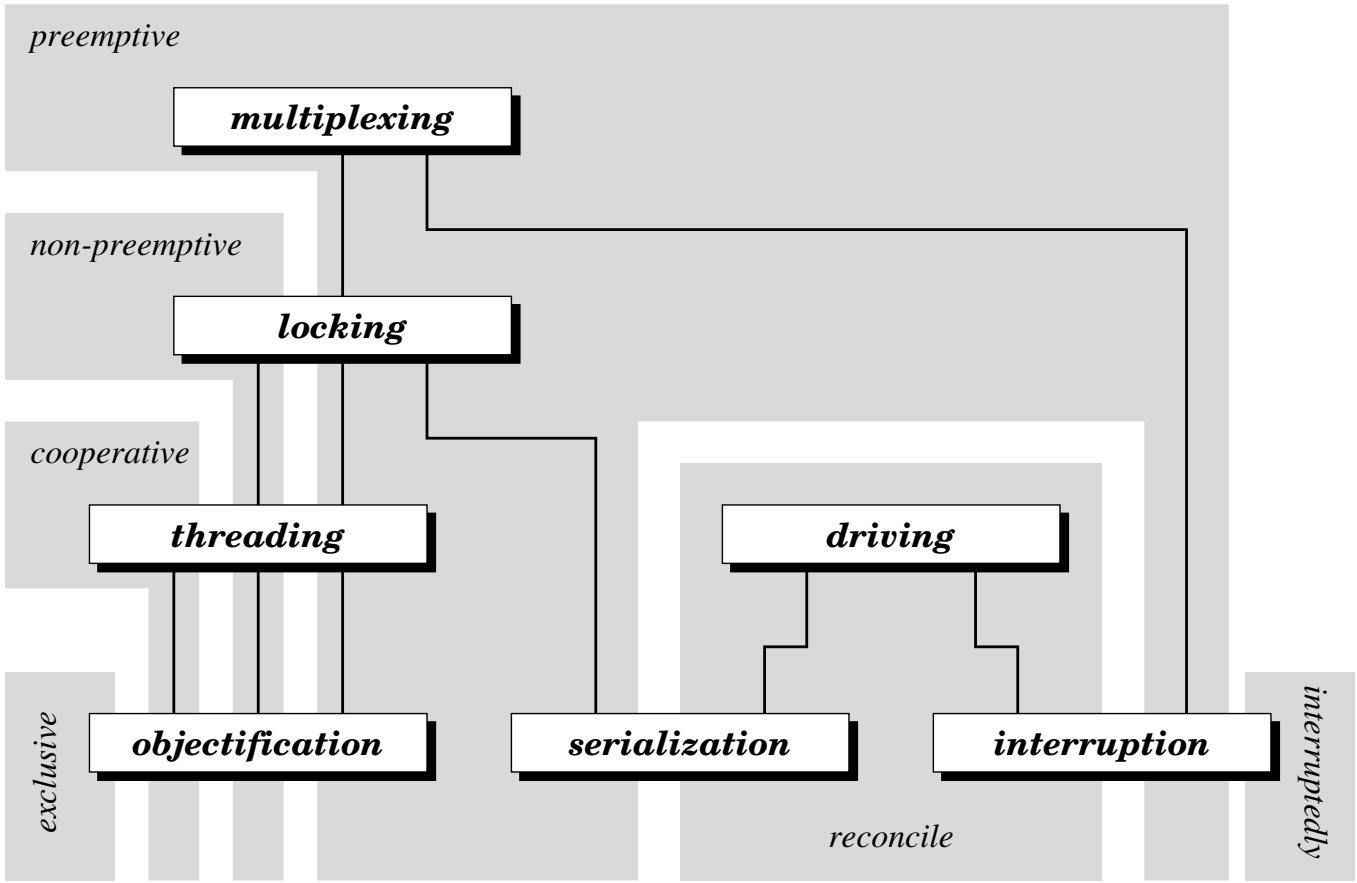
Ansatz: Programmfamilien im BS-Bau

- Parnas'79: „... users may demand that they are not be forced to pay for features that are unneeded.“

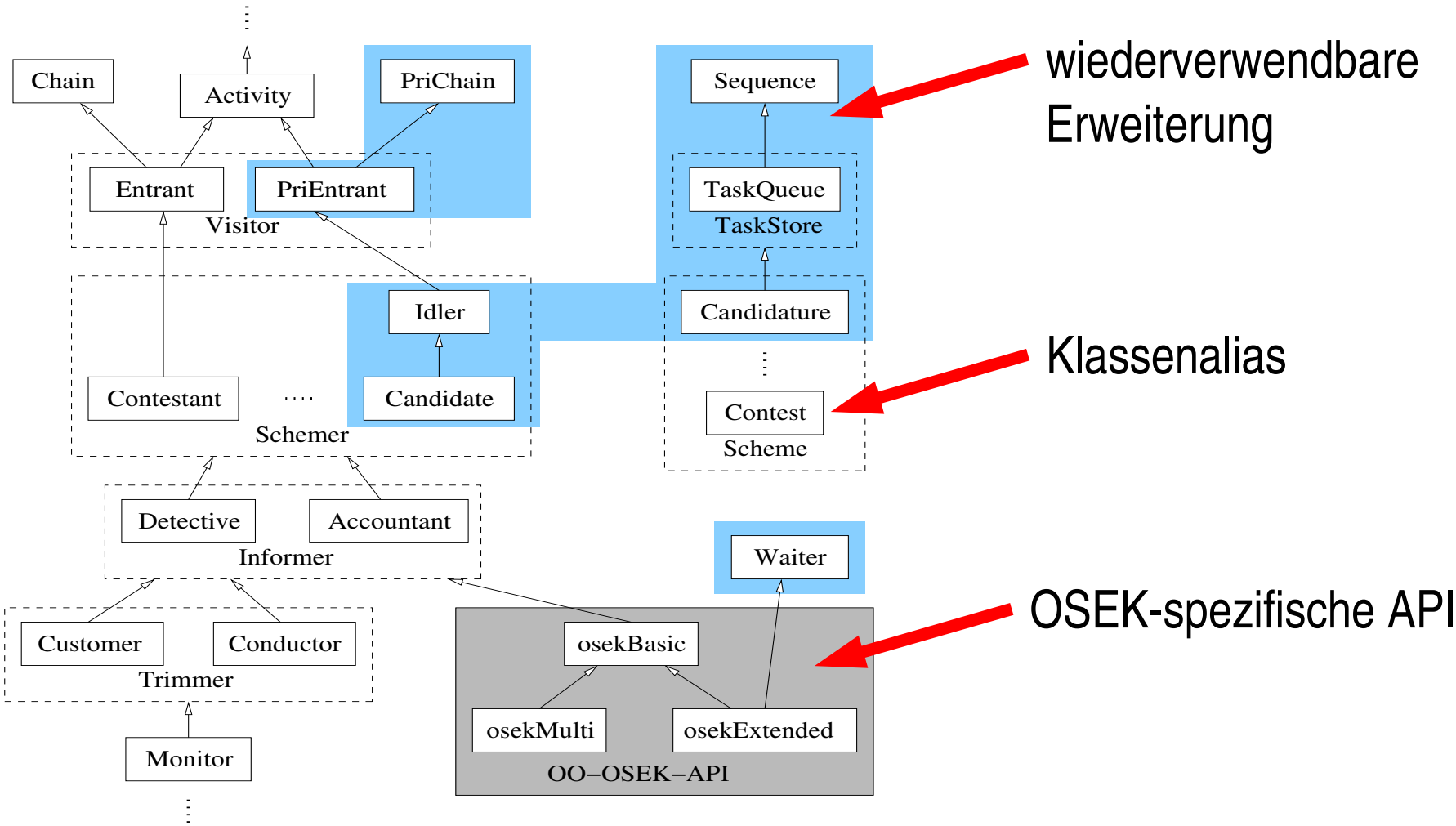


Ansatz: Die PURE Nucleus-Familie

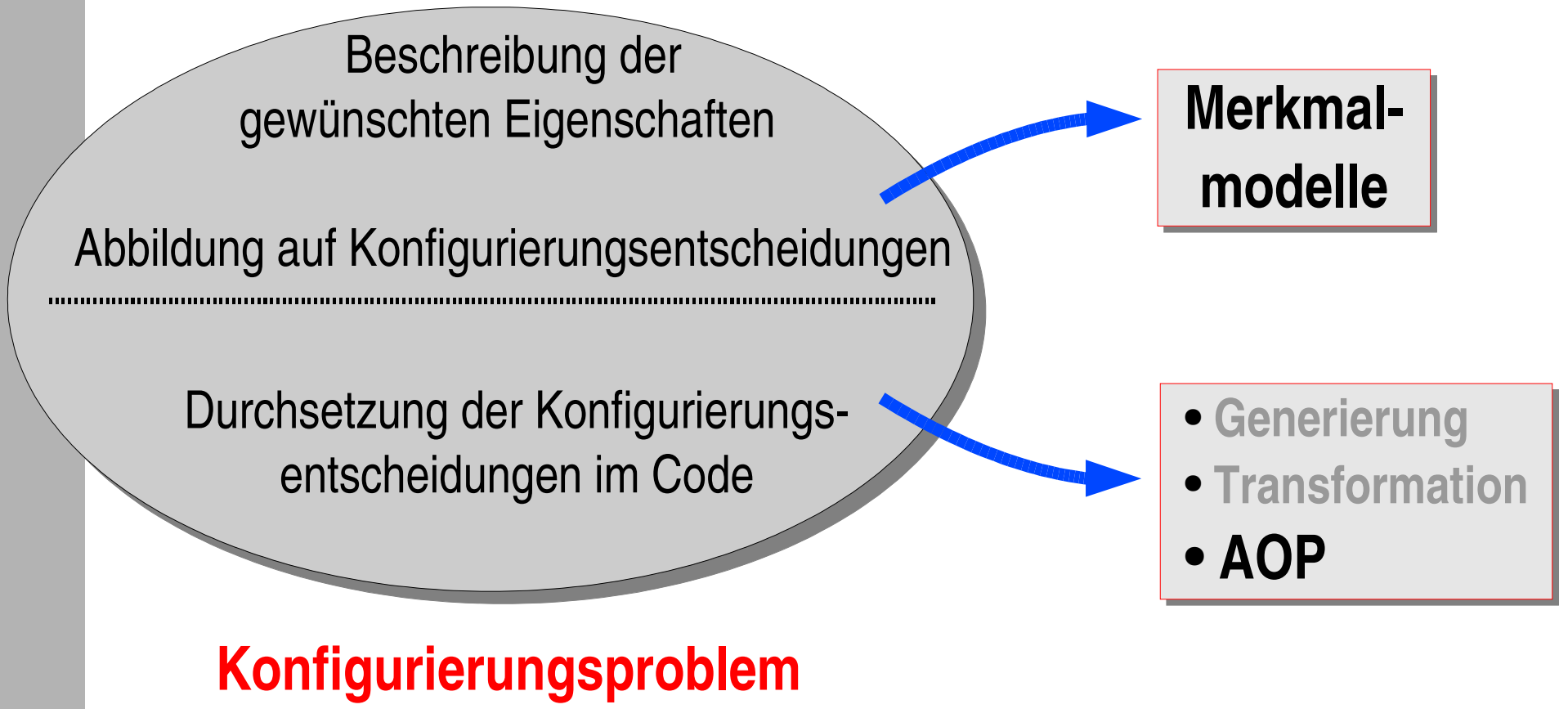
- Schröder-Preikschat'97



PURE/OSEK – '*threading*' Klassenstruktur

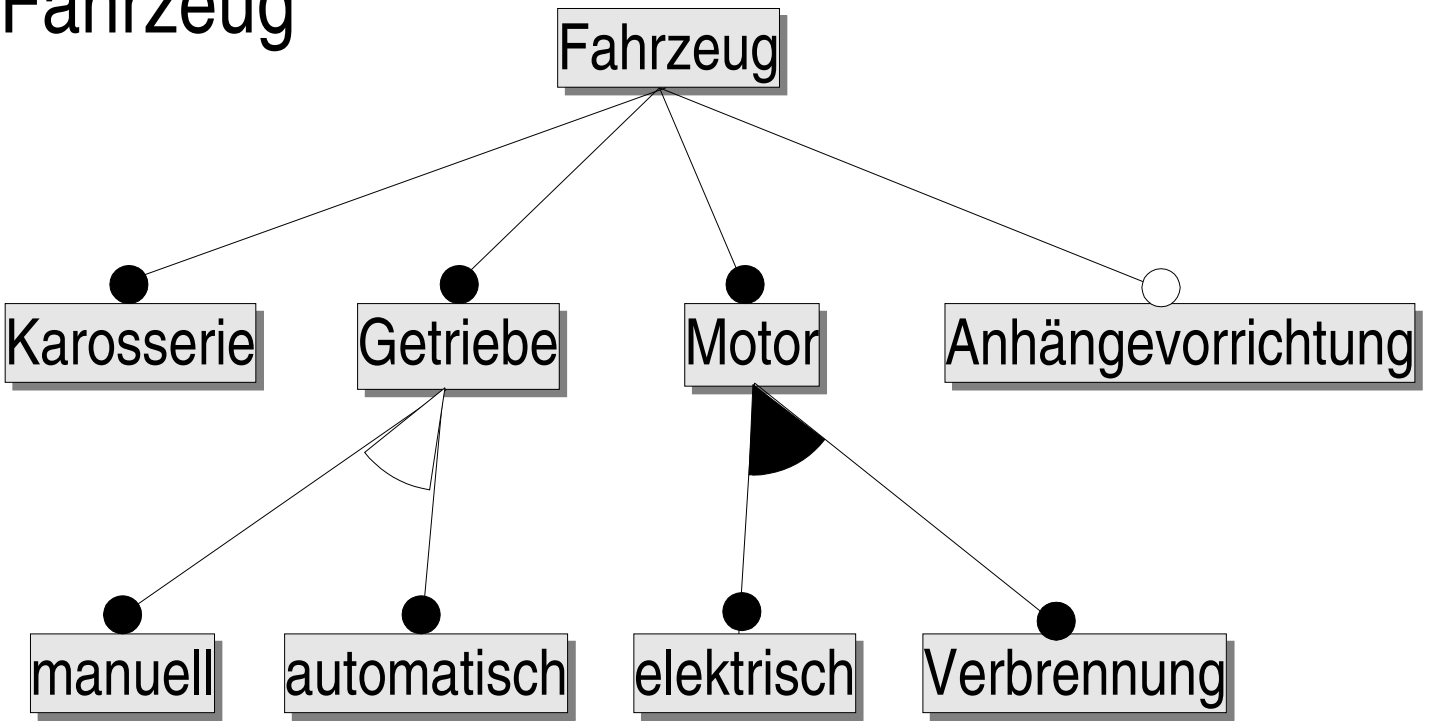


Modelle und Werkzeuge: Überblick



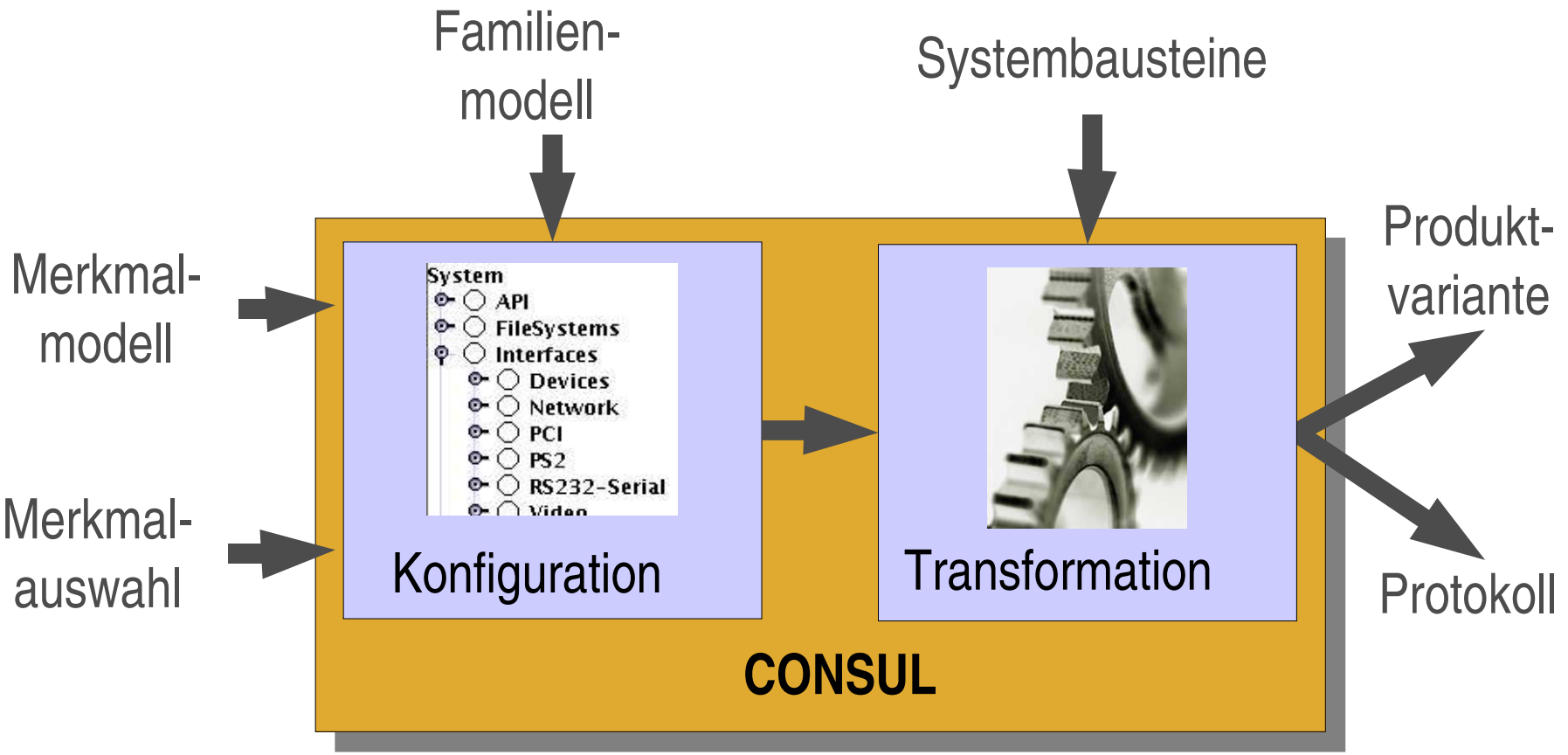
Merkmalmodelle (*feature models*)

Konzept: Fahrzeug*



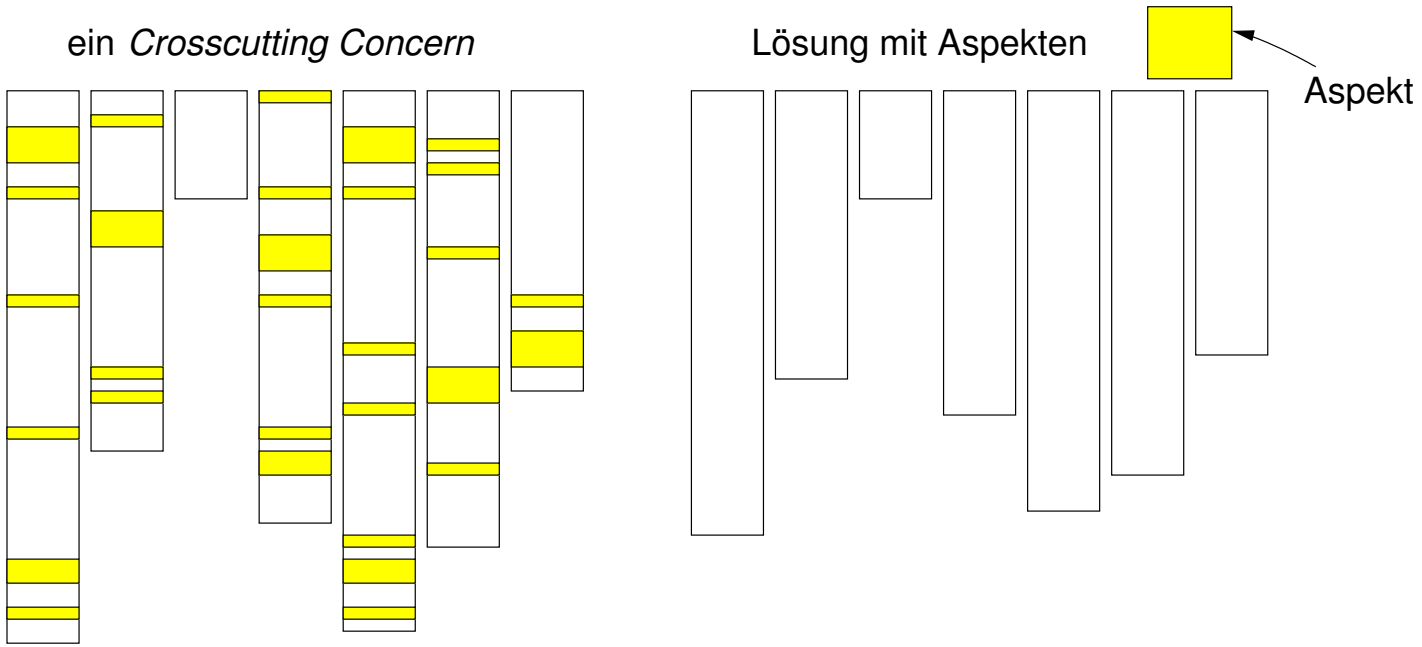
* Czarnecki, Eisenecker, ECOOP 99

PURE::Consul


Familien-
modell
Systembausteine
Merkmal-
modell
Merkmal-
auswahl
System
API
FileSystems
Interfaces
Devices
Network
PCI
PS2
RS232-Serial
Konfiguration
Transformation
Produkt-
variante
Protokoll
CONSUL

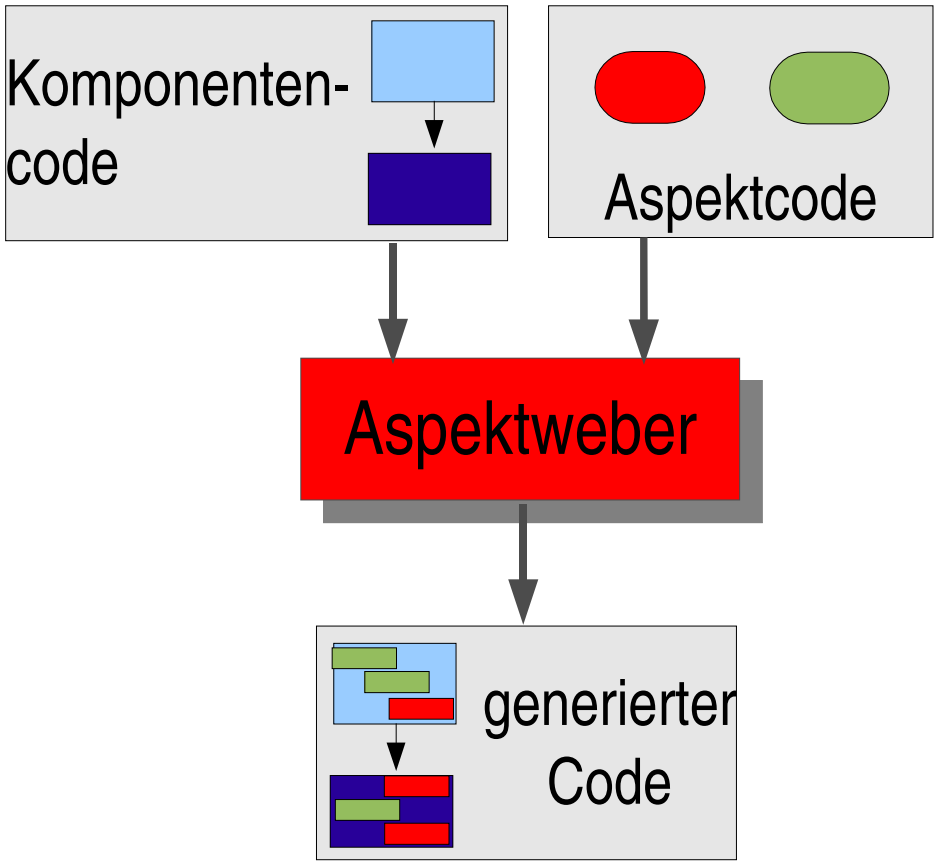
Aspektorientierte Programmierung (1)

- Kiczales'97: *Aspect-Oriented Programming* (AOP)
- Sprachlicher Ansatz, Ziel: **modulare** Implementierung von Querschnittsbelangen (*crosscutting concerns*)



Aspektorientierte Programmierung (2)

- Der **Aspektweber** verbindet Aspekt- und Komponentencode
- **Aspekte** legen die **Verbindungspunkte** fest



AspectC++

- AOP-Erweiterung für C++: **www.aspectc.org**
- Beispiel:

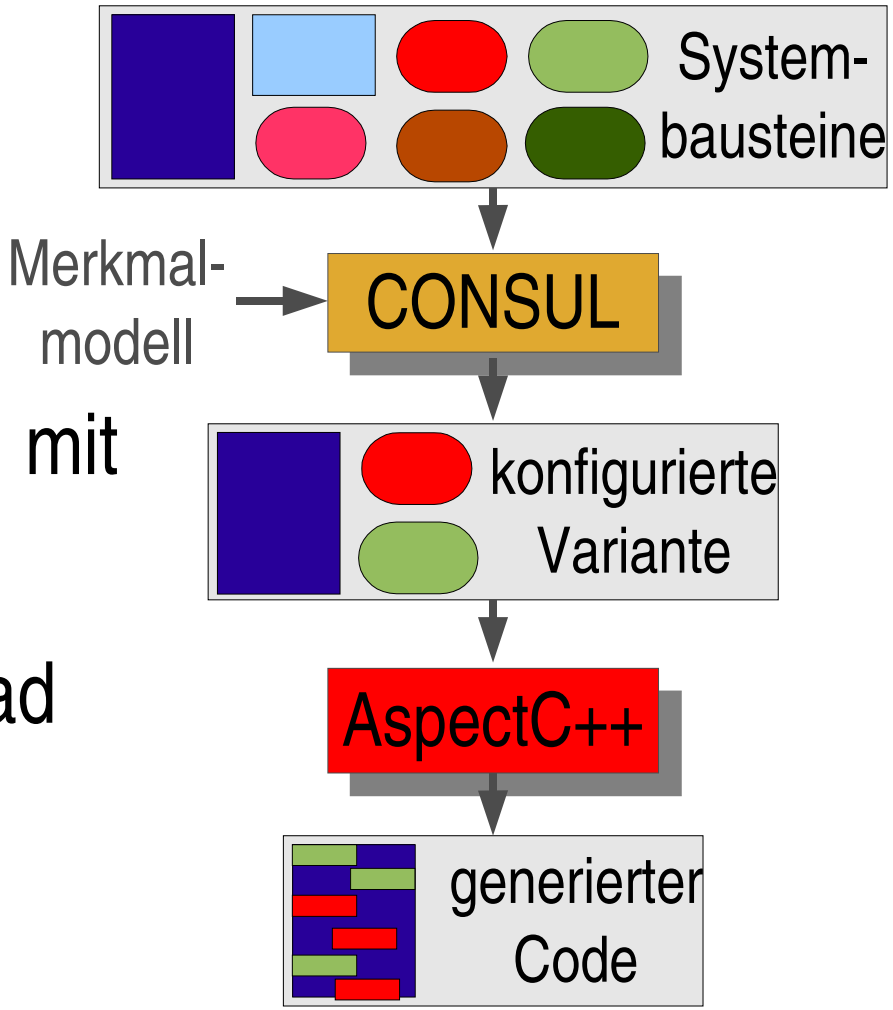
```
aspect FineGrainedInterruptSync {

  pointcut byte_xfer() =
    execution("void Bus::send(char)") ||
    execution("char Bus::receive()");

  advice byte_xfer() : around () {
    disable_int();
    tjp->proceed();
    enable_int();
  }
};
```

PURE::Consul ergänzt durch AspectC++

- weniger Schalter im Code
- einfachere Abbildung von Merkmalen auf Bausteine
- Konfigurierungsmöglichkeiten mit strategischem Charakter
- Hoher Wiederverwendungsgrad



Ergebnisse: Programmfamilienansatz

- Ressourcenverbrauch ...
  - skaliert
  - ist konkurrenzfähig (typ. OSEK Varianten: 3-4 KB)

Szenario	Größe*
Exclusive	434
Interruptedly	1268
Cooperative	1648
Non-preemptive	1699
Reconcile	2306
Preemptive	3078

* PURE/x86 in Bytes

Ergebnisse: Modelle und Werkzeuge

- Feature Modelle/PURE::Consul
  - Beherrschung von 321 C++ Klassen, 990 Quellcodedateien und 250 Features in PURE/OSEK
- AOP/AspectC++
  - Reduzierung der Konfigurationspunkte: Z.B. 166 Aufrufe in 16 Klassen → 1 konfigurierbarer Aspekt
- → Gute Voraussetzungen für komplexe Produktlinienentwicklungen