

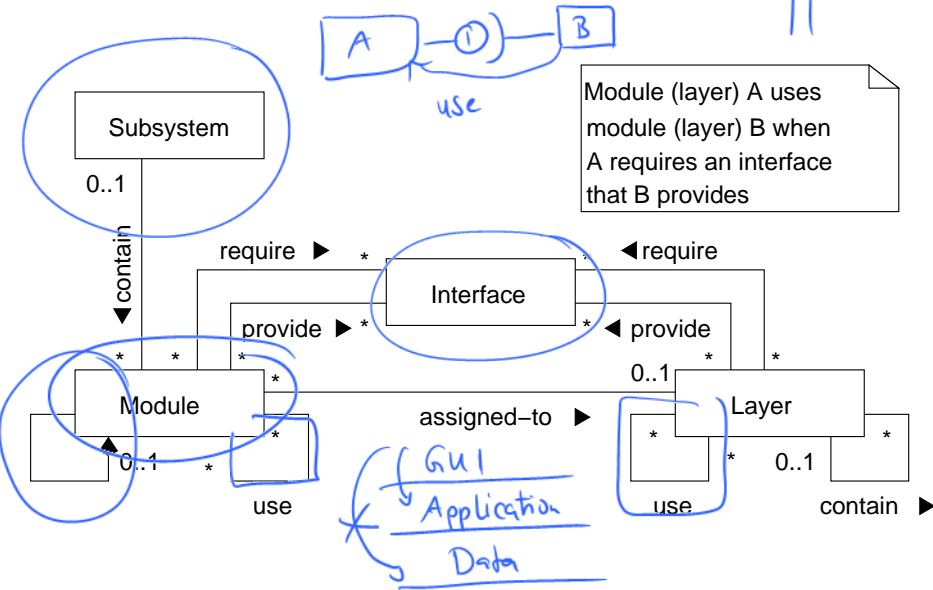
- bildet Komponenten und Konnektoren auf Module und Subsysteme ab
- setzt konzeptionelle Lösung mit aktuellen Softwareplattformen (Programmiersprachen, Bibliotheken) und Technologien um
- beschreibt statische Aspekte

- bildet Komponenten und Konnektoren auf Module und Subsysteme ab
- setzt konzeptionelle Lösung mit aktuellen Softwareplattformen (Programmiersprachen, Bibliotheken) und Technologien um
- beschreibt statische Aspekte

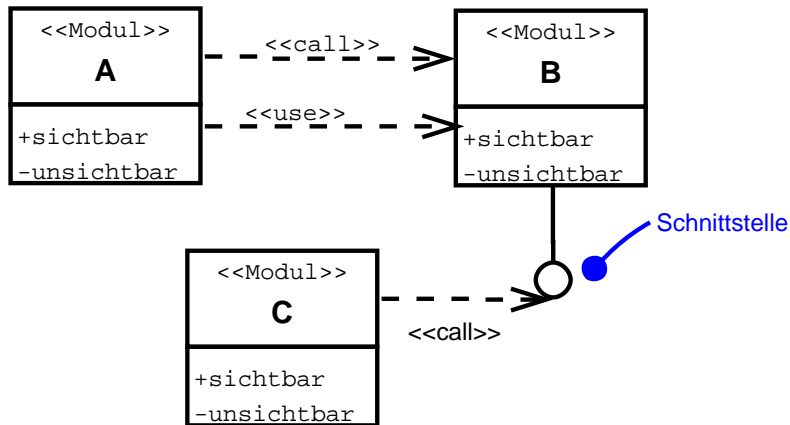
Engineering-Belange:

- Wie wird das Produkt auf die Software-Plattform abgebildet?
- Welche Softwaredienste werden benutzt und von wem?
- Wie wird das Testen unterstützt?
- Wie können Abhängigkeiten zwischen Modulen minimiert werden?
- Wie kann die Wiederverwendung von Modulen maximiert werden?
- Welche Techniken können eingesetzt werden, um Auswirkungen von Änderungen zu minimieren.

Modulblickwinkel (Hofmeister u. a. 2000)

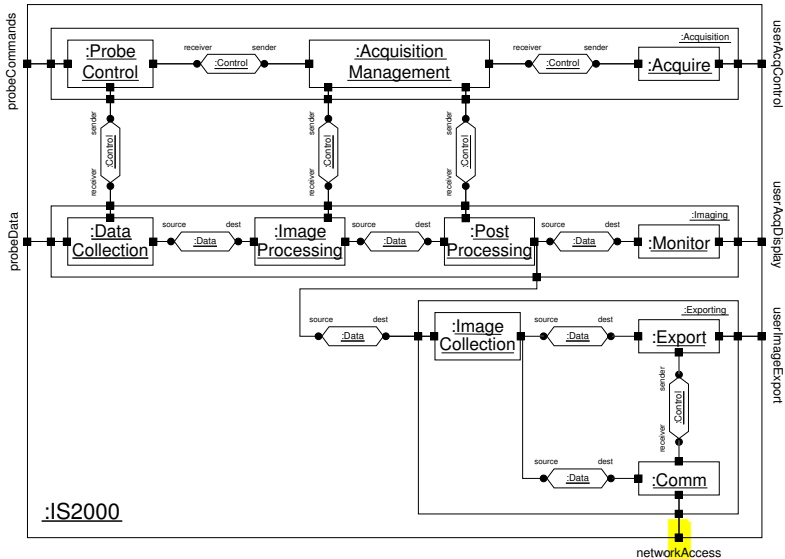


Notation für Module

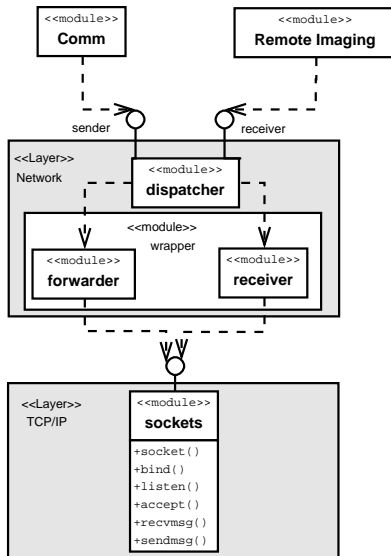


Strategie

Kapselung domänenspezifischer Bild-Daten.



Beispiel einer Modulsicht





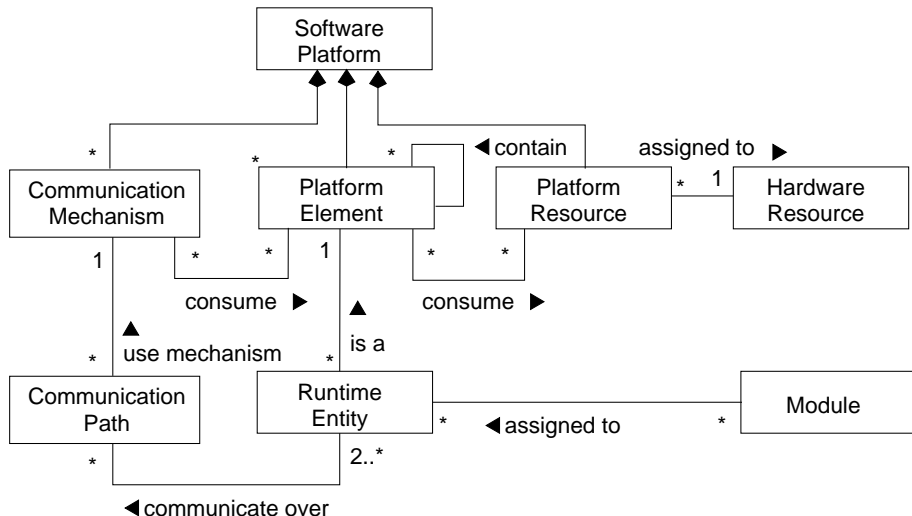
- beschreibt dynamische Aspekte
- beschreibt, wie Module auf Elemente der Ausführungs- und Hardwareplattform abgebildet werden
- definiert Laufzeitelemente und deren Attribute (Speicher- und Zeitbedarf, Allokation auf Hardware)

- beschreibt dynamische Aspekte
- beschreibt, wie Module auf Elemente der Ausführungs- und Hardwareplattform abgebildet werden
- definiert Laufzeitelemente und deren Attribute (Speicher- und Zeitbedarf, Allokation auf Hardware)

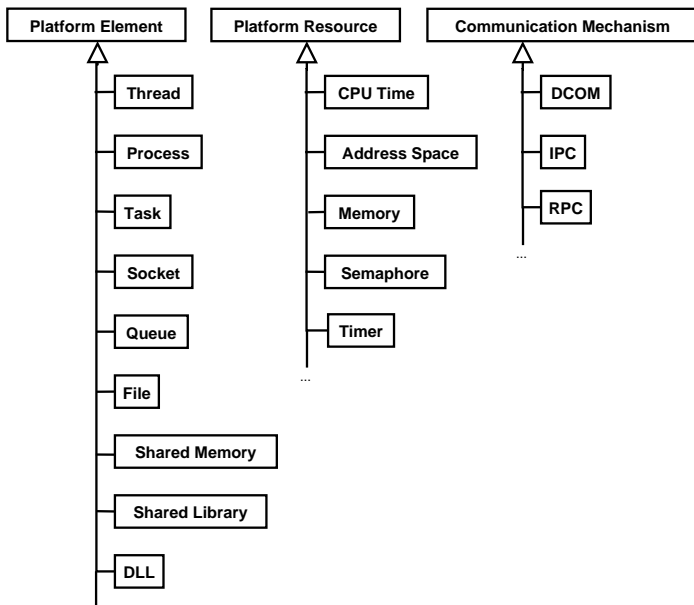
Engineering-Belange:

- Wie verläuft Kontroll- und Datenfluss zwischen Laufzeitkomponenten?
- Wie kann Ressourcenverbrauch ausgewogen werden?
- Wie können Nebenläufigkeit, Replikation und Verteilung erreicht werden, ohne die Algorithmen unnötig zu verkomplizieren?
- Wie können Auswirkungen von Änderungen in der Ausführungsplattform minimiert werden?

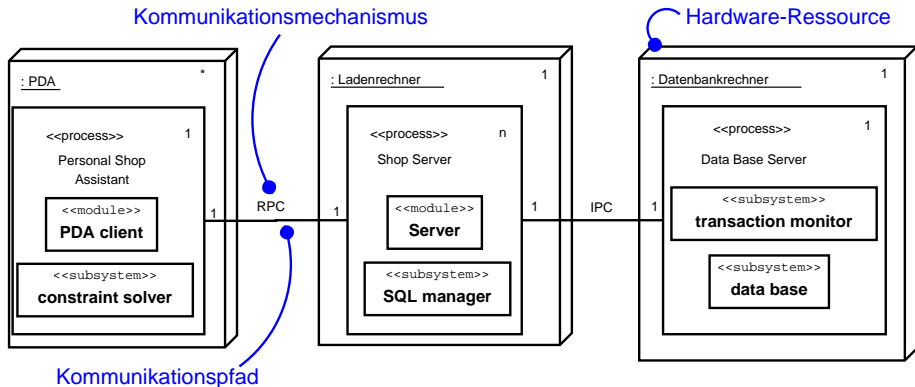
Ausführungsblickwinkel (Hofmeister u. a. 2000)

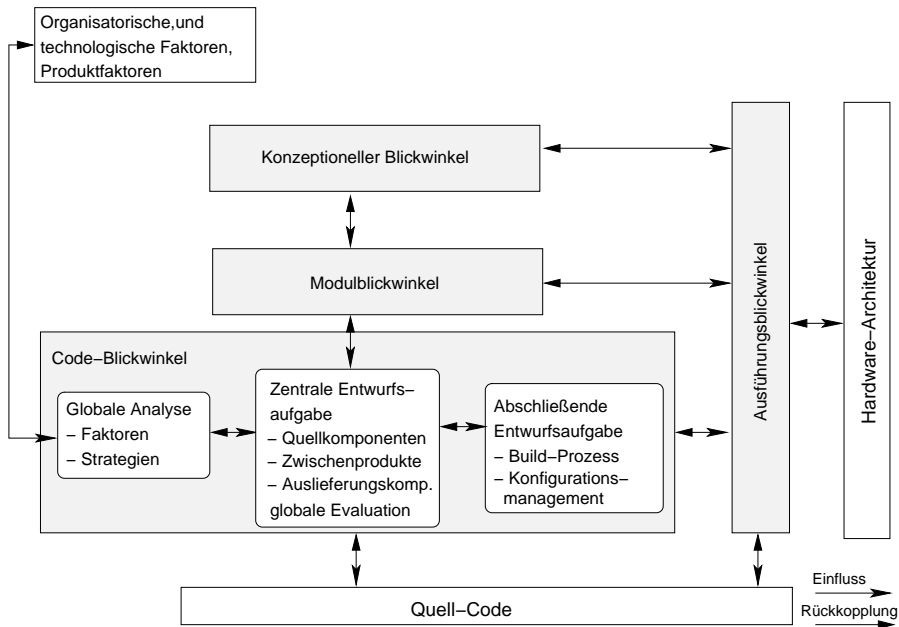


Ausführungsblickwinkel (Hofmeister u. a. 2000)



Beispiel einer Ausführungssicht





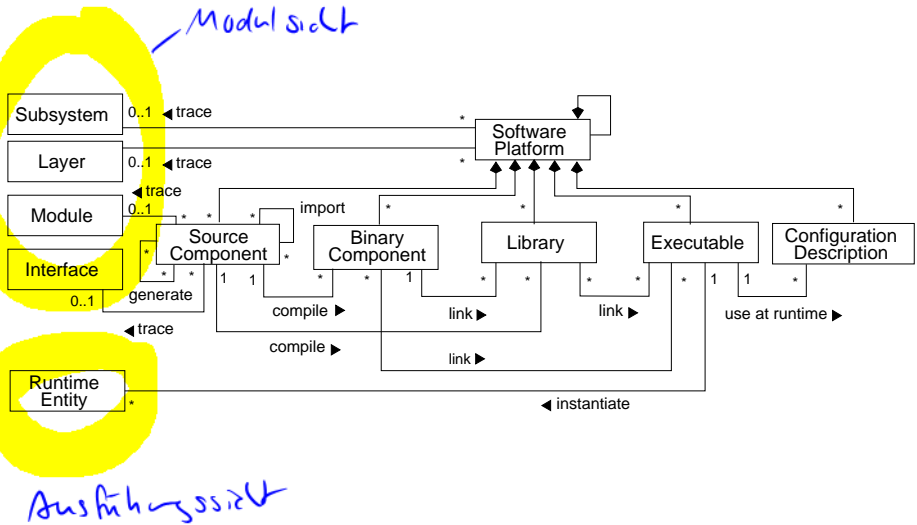
- bildet Laufzeiteinheiten auf installierbare Objekte (ausführbare Dateien, dynamische Link-Bibliotheken etc.) ab
- bildet Module auf Quellkomponenten (Dateien) ab
- zeigt, wie die installierbaren Dateien aus den Quellkomponenten generiert werden

- bildet Laufzeiteinheiten auf installierbare Objekte (ausführbare Dateien, dynamische Link-Bibliotheken etc.) ab
- bildet Module auf Quellkomponenten (Dateien) ab
- zeigt, wie die installierbaren Dateien aus den Quellkomponenten generiert werden

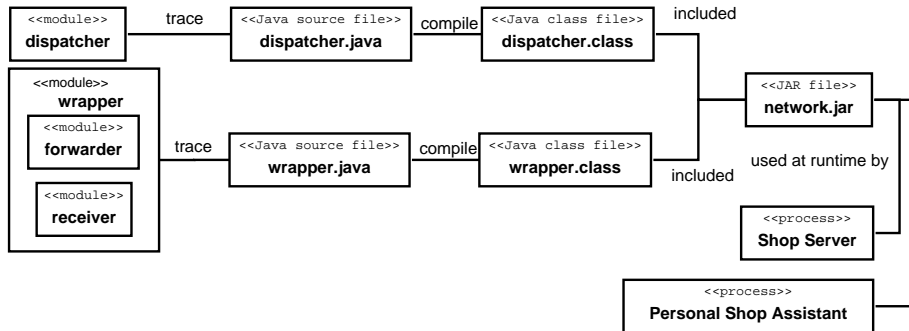
Engineering-Belange:

- Wie kann die Zeit und der Aufwand für Produkt-Upgrades verringert werden?
- Wie sollen Produktversionen verwaltet werden?
- Wie kann die Build-Zeit verringert werden?
- Welche Werkzeuge werden in der Entwicklungsumgebung benötigt?
- Wie wird Integration und Test unterstützt?

Code-Blickwinkel (Hofmeister u. a. 2000)



Code-Blickwinkel: Beispiel



Tabellendarstellung:

Module	Source File	JAR
dispatcher	dispatcher.java	network.jar
wrapper	wrapper.java	network.jar
...
JAR	used at runtime by	
network.jar	Shop Server, Personal Shop Assistant	
...	...	

Was ist Modularisierung?

Definition

Modularisierung: Dekomposition eines Systems in Module.

Modul: austauschbares Programmteil, das eine geschlossene Funktionseinheit bildet.

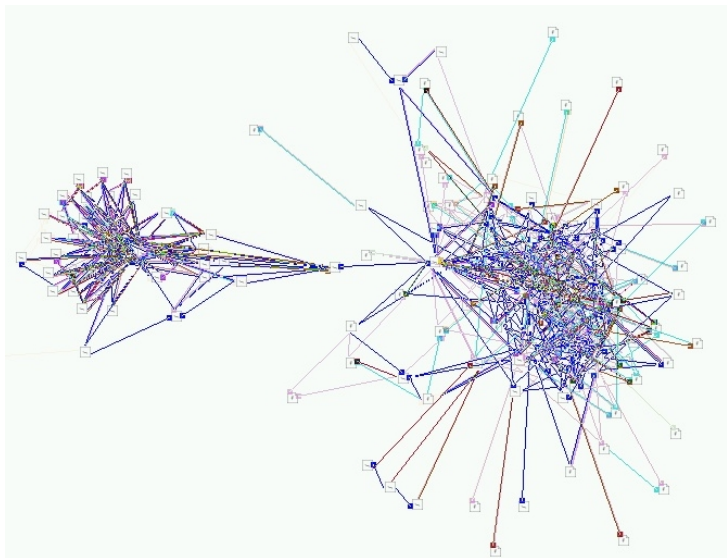
Arbeitspaket: von einer Person oder einer kleinen Gruppe von Entwicklern entwerfbar, implementierbar, testbar, verstehbar, änderbar, ...



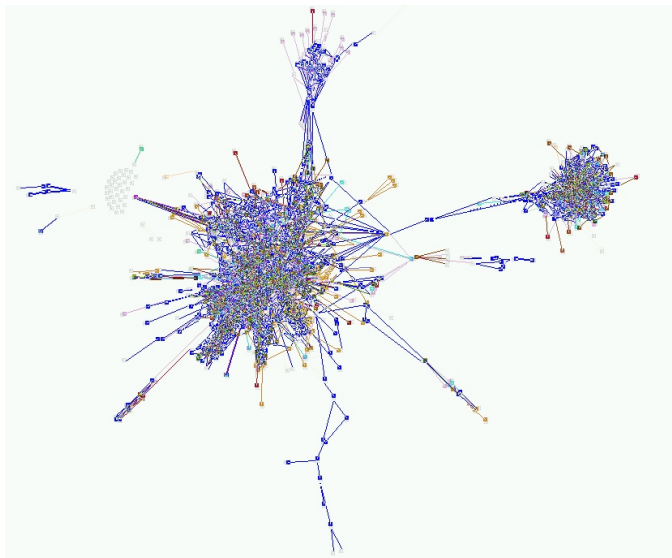
Schlechte Modularisierung



Modularisierung in der Praxis



Modularisierung in der Praxis



Modularisierung in der Praxis

