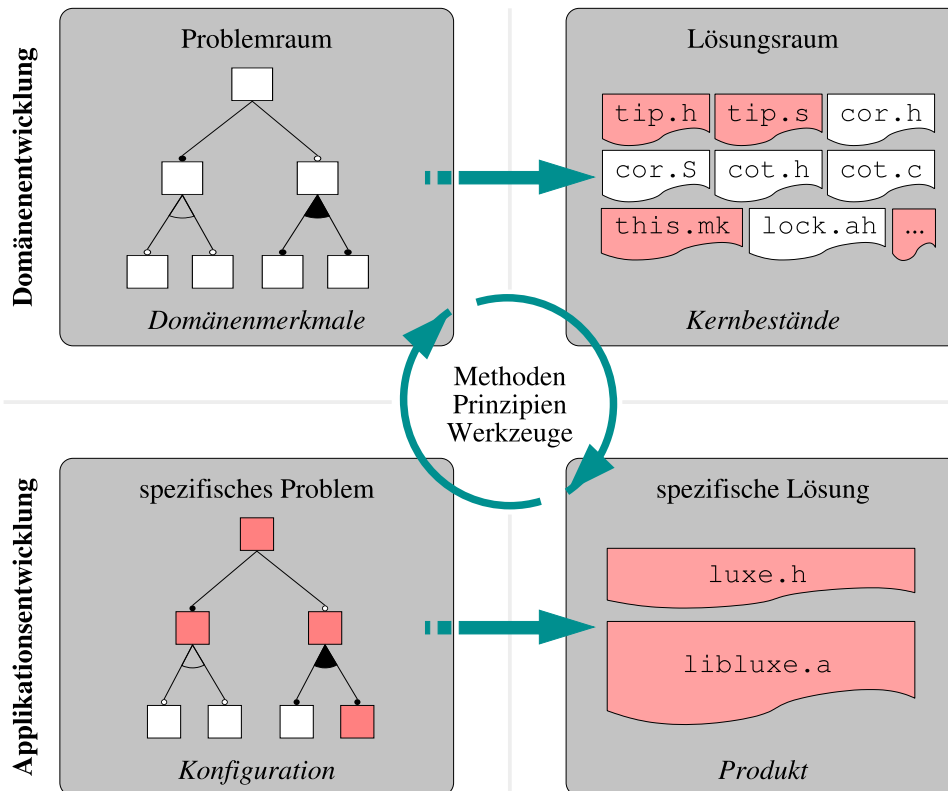


### 3 Übungsaufgabe #3: Merkmalmodellierung

In der Vorlesung wurde der Prozess der Softwareentwicklung mit Produktlinien vorgestellt. In dieser Aufgabe sollen Teile aus diesem Prozess eingeübt werden. Zur Erinnerung betrachten Sie bitte folgendes Diagramm und arbeiten Sie die vierte Vorlesung nochmals durch:



In dieser Aufgabe soll am Beispiel der in den Übungen entwickelten *abstrakten Maschine* die Trennung des *Problem-* vom *Lösungs-*raum verdeutlicht werden. Außerdem werden unterstützende Werkzeuge in den Übersetzungsprozess Ihrer Lösung integriert.

#### 3.1 Merkmalmodell für die Übungsaufgabe

In den Übungsaufgaben #1 und #2 wurden Sie an verschiedenen Stellen angehalten, verschiedene Varianten von Lösungen zu implementieren. Diese Variabilität soll nun in einem Merkmalmodell, wie sie in der Vorlesung vorgestellt wurden, beschrieben werden. Ein Merkmalmodell ist ein azyklischer Graph, in dem die folgenden Elemente erlaubt sind:

- zwingende Merkmale
- wahlfreie Merkmale
- Gruppen von Alternativen
- Gruppen von Anhäufungen, sowie
- weitere Einschränkungen und Abhängigkeiten

Die Umsetzung erfolgt in der Sprache des Linux Konfigurationswerkzeugs `Kconfig`.

---

### 3.2 Das Linux Konfigurationswerkzeug: `make xconfig`

Die Entwicklung mit Produktlinien sieht vor, dass auch nicht-Domänenexperten spezifische Lösungen generieren können. In Linux wurde dazu ein spezielles Werkzeug `Kconfig` entwickelt, welches es den Benutzern erlaubt, den Kern (graphisch) zu konfigurieren. Dieses Werkzeug wird in den Linux Quellen mit den Kommandos `make menuconfig` oder `make xconfig` aufgerufen. `Kconfig` stellt sicher, dass die vom Benutzer getroffene *Selektion* im Variantenmodell *valid* ist. Aus dieser Selektion werden dann einbindbare *Header*-Dateien und *Make*-Fragmente erzeugt, die Sie in Ihren Übersetzungsprozess integrieren können und sollen.

### 3.3 Merkmaldiagramm für die Fallstudie *Synchronisation*

In der Vorlesung wurde als Fallstudie zum Themenkomplex *Programmfamilien* Merkmaldiagramme zur Domäne *Synchronisation in Betriebssystemen* gezeigt. Modellieren sie daher das Merkmaldiagramm auf Folien 4–27 bis 4–30 mit dem Linux Konfigurationswerkzeug `Kconfig`. Bitte beachten Sie, dass in den Folien das Modell aus Platzgründen nicht vollständig sondern nur ausschnittsweise abgebildet wurde. Ihre Lösung sollte jedoch ganzheitlich umgesetzt sein.

#### Aufgaben:

- Modellieren Sie Ihr bisheriges Variantenmodell als Merkmaldiagramm.
- Implementieren Sie dieses Modell mit dem Linux-Konfigurationsprogramm `Kconfig`.
- Integrieren Sie das Linux-Konfigurationsprogramm in Ihre vorhandenen Makefiles. Es ist ihnen freigestellt ob Sie `Kconfig` aus den Quellen bauen oder vorübersetzte Programmdateien verwenden.
- Implementieren Sie das Merkmaldiagramm aus den Folien 4–27 bis 4–30 der Vorlesung in `Kconfig`.  
**NB:** in den Folien ist das Featuremodell nur ausschnittsweise implementiert. Für diese Aufgabe ist jedoch das Gesamtmodell umzusetzen.

#### Hinweise:

- Eine angepasste Version des Linux Konfigurationswerkzeug ist von der Website zu dieser Veranstaltung verlinkt. Diese Version basiert auf den Quellen aus dem Linux Baum Version 2.6.30-rc7, kann jedoch unabhängig von den restlichen Quellen des Linuxkerns übersetzt und benutzt werden.
- Eine bereits vorübersetzte ausführbare Version des Linux Konfigurationswerkzeugs `Kconfig` finden Sie im CIP-Pool unter `/proj/i4ose/tools/bst-kconfig`.
- Um den Konfigurator zu benutzen, starten Sie das Werkzeug mit dem Featuremodell als Programmparameter:  
`mconf features.fm`.
- Um aus einer Selektion (üblicherweise `.config`) einbindbare C-Header zu erzeugen, benutzen Sie das Programm `conf` mit einem speziellem Parameter: `conf -s features.fm`. Die von uns modifizierte Version erstellt darauf hin die Dateien `include/auto.conf` und `include/autoconf.h`.  
Mit den (Linux) Makedateien ist diese Funktionalität mit dem Maketarget `silentoldconfig` erreichbar.
- Die erzeugte `include/autoconf.h` Datei soll (genau wie in Linux auch) als sogenanntes *forced #include* eingebunden werden. Verwenden Sie hierzu die Übersetzeroption `-include` von `gcc`.
- Das Verzeichnis `include/` muss bereits existieren. Die `Kconfig` Programme erstellen es nicht von selber sondern brechen mit einem Fehler ab.

### 3.4 Abgabe: am 20.05.2010